



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
DE COMPUTAÇÃO



Traçamento de raios sísmicos com múltiplas condições iniciais usando Redes de Kolmogorov-Arnold com Informações Físicas

Lucas Torres Marques

Orientador: Prof. Dr. Tiago Tavares Leite Barros

Co-orientador: Prof. Dr. Luiz Affonso Henderson Guedes de Oliveira

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação da UFRN (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Mestre em Ciências.

Natal, RN, dezembro de 2025

Universidade Federal do Rio Grande do Norte - UFRN
Sistema de Bibliotecas - SISBI
Catalogação de Publicação na Fonte. UFRN - Biblioteca Central Zila Mamede

Marques, Lucas Torres.

Traçamento de raios sísmicos com múltiplas condições iniciais usando Redes de Kolmogorov-Arnold com Informações Físicas / Lucas Torres Marques. - 2025.
80f.: il.

Dissertação (Mestrado) - Universidade Federal do Rio Grande do Norte, Centro de Tecnologia, Programa de Pós Graduação em Engenharia Elétrica e de Computação, Natal, 2025.

Orientação: Dr. Tiago Tavares Leite Barros.

Coorientação: Dr. Luiz Affonso Henderson Guedes de Oliveira.

1. Traçamento de Raios Sísmicos - Dissertação. 2. Redes de Kolmogorov-Arnold - Dissertação. 3. Redes Neurais com Informação Física - Dissertação. 4. Geofísica Computacional - Dissertação. 5. Métodos de Aprendizado de Máquina - Dissertação. I. Barros, Tiago Tavares Leite. II. Oliveira, Luiz Affonso Henderson Guedes de. III. Título.

RN/UF/BCZM

CDU 004

*À minha esposa, que fez do amor um
lugar seguro durante esta jornada.
Obrigado por permanecer ao meu
lado em cada etapa, sem condições,
sem reservas.*

Agradecimentos

Ao meu orientador e ao meu co-orientador, professores Dr. Tiago Barros e Dr. Luiz Affonso Guedes, pela orientação cuidadosa, pela disponibilidade ao longo de todo o mestrado e pelas contribuições decisivas para o desenvolvimento deste trabalho. Mais do que guias acadêmicos, foram mestres no sentido mais pleno da palavra.

Ao colega pesquisador Dr. Ramon Araujo e ao professor Dr. Ivanovitch Silva, pela ajuda na revisão deste trabalho e pelas observações que contribuíram para aprimorar a apresentação final do texto.

Aos colegas do laboratório Model2AI, pelas sugestões, pelas discussões e pela troca de experiências que enriqueceram esta pesquisa.

Aos demais colegas de pós-graduação pelas críticas construtivas, pelo convívio e pelo ambiente de aprendizado compartilhado ao longo desta jornada, destacando, em especial, meus amigos Ricardo, Nathan e Leandro.

Aos amigos que estiveram presentes ao longo desta jornada e que, com apoio, escuta e incentivo, tornaram o percurso do mestrado mais leve e possível.

À minha família, agradeço pelo apoio constante, pela compreensão e pelo incentivo nos momentos mais exigentes do percurso.

À minha esposa, pelo amor, paciência e apoio constantes ao longo desta caminhada.

Este trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001

Resumo

A modelagem de propagação de ondas sísmicas é fundamental para o imageamento sísmico e a caracterização de reservatórios na indústria de petróleo e gás. O traçamento de raios sísmicos é uma técnica eficiente, mas métodos tradicionais enfrentam dificuldades em modelos geológicos complexos devido à instabilidade e ao alto custo computacional, uma vez que exigem a resolução de uma única condição inicial por vez. Este trabalho propõe o uso de Redes de Kolmogorov-Arnold com Informação Física (PIKANs, do inglês *Physics-Informed Kolmogorov-Arnold Networks*) para superar essas limitações, permitindo o traçamento simultâneo de múltiplos raios com diferentes condições iniciais. A metodologia inclui a suavização do modelo de velocidade com filtro de B-splines, a utilização de uma função de perda composta e o uso de pesos adaptativos na função de perda física para melhorar o treinamento do modelo. Foram realizados dois experimentos com dados sintéticos: o primeiro com um modelo de velocidade parametrizado e o segundo com um modelo não parametrizado usado como referência na área de processamento sísmico, chamado de Marmousi. Os resultados demonstram a consistência da abordagem, com valores de R^2 próximos a 1 para o primeiro experimento e superiores a 0.90 para o segundo. De forma qualitativa, o método traça raios sísmicos com precisão, inclusive em regiões não utilizadas no treinamento e em áreas com rápida variação espacial de velocidade, mostrando-se mais preciso em prever as coordenadas espaciais do raio do que o vetor de vagariedade.

Palavras-chave: Traçamento de Raios Sísmicos, Redes de Kolmogorov-Arnold, Redes Neurais com Informação Física, Geofísica Computacional, Métodos de Aprendizado de Máquina.

Abstract

Seismic wave propagation modeling is fundamental for seismic imaging and reservoir characterization in the oil and gas industry. While ray tracing is an efficient technique, traditional methods face challenges in complex geological models due to instability and high computational cost, as they require solving for a single initial condition at a time. This work proposes the use of Physics-Informed Kolmogorov-Arnold Networks (PIKANs) to overcome these limitations by allowing the simultaneous tracing of multiple rays with different initial conditions. The proposed methodology includes velocity model smoothing with B-spline filters, the use of a compound loss function, and the application of adaptive weights in the physics-informed loss function to enhance model training. Two experiments were conducted using synthetic data: the first with a parameterized velocity model, and the second with a non-parameterized model widely used as a benchmark in seismic processing, known as the Marmousi model. The results demonstrate the consistency of the approach, with R^2 values close to 1 for the first experiment and exceeding 0.90 for the second. Qualitatively, the method accurately traces seismic rays, even in regions not used for training and in areas with rapid spatial velocity variation, proving to be more precise at predicting the ray's spatial coordinates than the slowness vector.

Keywords: Seismic Ray Tracing, Kolmogorov-Arnold Networks, Physics-Informed Neural Networks, Computational Geophysics, Machine Learning Methods.

Lista de Figuras

2.1	Trajetória de um raio sísmico no plano $x - z$, do ponto inicial (x_0, z_0) na região W_1 ao ponto final (x^*, z^*) na região W_2 , com ângulos inicial e final θ_0 e θ_f , respectivamente.	5
2.2	Diagrama representando o funcionamento das Redes Neurais com Informação Física (PINNs).	7
2.3	Arquitetura simplificada da KAN, onde as entradas são as variáveis x e y , e a saída z está relacionada à estimativa da função desejada, conforme a estrutura da rede.	9
2.4	Exemplo das funções de base B-splines (B_k) após a expansão da grade para $G = 5$ e com splines de ordem $k = 3$	10
2.5	Exemplo da aplicação de um <i>mirror padding</i> em um conjunto de dados bidimensional, como o campo de velocidade. Em vermelho estão os valores do conjunto de dados original, enquanto nos azuis estão os dados espelhados.	13
4.1	Esquema geral da nova abordagem proposta para o traçamento de raios sísmicos. Processos são representados em preto, entradas dos processos e hiperparâmetros em verde, e saídas em laranja.	20
4.2	Divisão do Conjunto de Dados original em treinamento, validação e teste.	29
5.1	Raios traçados para o modelo de velocidade parametrizado.	49
5.2	Impacto do tamanho da grade nas métricas de avaliação do modelo PIKAN para o modelo de velocidade parametrizado. As barras coloridas representam as métricas para cada variável alvo, enquanto a linha preta indica a média do modelo.	51
5.3	Evolução das componentes da função de perda durante o treinamento do modelo PIKAN no modelo de velocidade parametrizado. Painel superior: perda total de treinamento. Painel central: perda física de treinamento. Painel inferior: comparação entre as perdas de dados nos conjuntos de treinamento e validação.	54
5.4	Comparação visual entre trajetórias de raios sísmicos calculadas pelo método de Runge-Kutta de segunda ordem (linhas sólidas cinzas) e previstas pelo modelo PIKAN (linhas tracejadas pretas). O campo de velocidade parametrizado é representado pelo gradiente de cores, com valores variando de 1,5 km/s (azul) a 4,5 km/s (vermelho).	56
5.5	Raios traçados para o modelo de velocidade do Marmousi a partir do método Runge-Kutta de segunda ordem.	59

5.6	Comparação visual do Modelo de Velocidade do Marmousi Original (painel superior esquerdo) com as versões suavizadas por interpolação B-splines, variando o fator de subamostragem (s). Valores baixos ($s = 20$ e $s = 25$) preservam contornos e heterogeneidades, mas podem reter descontinuidades locais; $s = 30$ mantém interfaces principais com redução efetiva de variações espúrias; valores altos ($s = 35$ e $s = 40$) promovem suavização excessiva, com perda visível de detalhes finos e contraste entre camadas.	61
5.7	Avaliação quantitativa do desempenho do modelo PIKAN em função do fator de subamostragem (s) no conjunto de validação.	62
5.8	Avaliação quantitativa do desempenho do modelo PIKAN em função do tamanho de grade (G) no conjunto de validação.	63
5.9	Evolução das componentes da função de perda durante o treinamento do modelo PIKAN no modelo de velocidade do Marmousi. Painel superior: perda total de treinamento. Painel central: perda física de treinamento. Painel inferior: comparação entre as perdas de dados nos conjuntos de treinamento e validação.	65
5.10	Comparação visual entre trajetórias de raios sísmicos calculadas pelo método de Runge-Kutta de segunda ordem (linhas sólidas cinzas) e previstas pelo modelo PIKAN (linhas sólidas pretas). O campo de velocidade parametrizado é representado pelo gradiente de cores, com valores variando de 1,5 km/s (roxo) a 6 km/s (amarelo).	67

Lista de Tabelas

5.1	Exemplo das cinco primeiras amostras do conjunto de dados de treinamento para o modelo de velocidade parametrizado.	48
5.2	Métricas de avaliação para as predições do PIKAN no traçamento de raios sísmicos.	55
5.3	Exemplo de cinco amostras escolhidas aleatoriamente do conjunto de dados de treinamento para o modelo de velocidade do Marmousi.	59
5.4	Resultados obtidos para o traçamento de raios com PIKAN no modelo de velocidade do Marmousi.	66
5.5	Comparação quantitativa entre PIKAN (KAN) e PINN tradicional (MLP) no conjunto de teste do Marmousi.	68

Lista de Símbolos e Abreviaturas

E	Número total de arestas (funções de ativação em cada conexão)
G	Número de intervalos da grade em cada splines (tamanho da grade)
N	Distribuição normal
R^2	<i>R-squared</i> ou Coeficiente de Determinação
Φ	Função de ativação externa da Rede de Komogorov-Arnold
$\beta(\cdot)$	Função de base B-splines
$\beta^3(\cdot)$	Função de base B-splines cúbica
Φ	Camada qualquer de uma Rede de Komogorov-Arnold
ε	Parâmetro de tolerância de valor muito pequeno
η	Taxa de aprendizagem
λ_d	Pesos da Função de Perda dos Dados
λ_f	Pesos da Função de Perda Física
λ_{reg}	Pesos da Função de Perda que associa todos os termos de regularização
$\mathbf{p}$	Vetor de vagariedade no plano xz , onde $\mathbf{p} = (p_x, p_z)$
$\mathbf{r}$	Vetor de estado do raio
$\mathbf{x}$	Vetor de posição no plano xz , onde $\mathbf{x} = (x, z)$
$\mathcal{L}_{\text{reg}}$	Função de Perda que associa todos os termos de regularização utilizados na implementação das Redes de Kolmogorov-Arnold
$\mathcal{L}_{\text{total}}$	Função de Perda Composta
$\mathcal{L}_d$	Função de Perda dos Dados
$\mathcal{L}_f$	Função de Perda Física
$\mathcal{N}[\cdot]$	Operador diferencial

ϕ	Função de ativação interna da Rede de Komogorov-Arnold
$\rho(q)$	Frequência de dados para cada quadrante do modelo de velocidade
dg	Espaçamento da grade. Distância entre os coeficientes B-splines do modelo de velocidade
dg_{over}	Espaçamento entre pontos adjacentes do campo de velocidade discretizado
dt	Passo de integração
dx	Incremento de distância na dimensão x
dz	Incremento de distância na dimensão z
θ_0	Ângulo de disparo
$c[i, j]$	Coeficientes B-splines do modelo de velocidade
k	Ordem ou grau das B-splines
s	Fator de subamostragem
t_{max}	Tempo máximo de simulação
$v(\mathbf{x})$	Velocidade de propagação da onda no ponto $\mathbf{x}$
w_f	Pesos adaptativos da Função de Perda Física
x_{range}	Intervalo de valores para a coordenada espacial x
z_{range}	Intervalo de valores para a coordenada espacial z
AD	<i>Automatic Differentiation</i>
EDO	Equação Diferencial Ordinária
EDP	Equação Diferencial Parcial
IIR	<i>Infinite Impulse Response</i>
KAN	<i>Kolmogorov-Arnold Network</i>
MAE	<i>Mean Absolute Error</i>
MAPE	<i>Mean Absolute Percentage Error</i>
MLP	<i>Multilayer Perceptron</i>
MSE	<i>Mean Squared Error</i>
PIKAN	<i>Physics-Informed Kolmogorov-Arnold Network</i>

PINN *Physics-Informed Neural Network*

RK Runge-Kutta

RK2 Runge-Kutta de segunda ordem

RMSE *Root Mean Squared Error*

Sumário

Lista de Figuras	i
Lista de Tabelas	iii
Lista de Símbolos e Abreviaturas	iv
Sumário	vi
1 Introdução	1
1.1 Hipótese Científica e Objetivo Geral da Dissertação	2
1.2 Objetivos Específicos	3
1.3 Estrutura do texto	3
2 Fundamentação Teórica	4
2.1 Traçamento de Raios Sísmicos	4
2.2 Redes Neurais com Informação Física (PINNs)	6
2.3 Redes de Kolmogorov-Arnold (KAN)	8
2.3.1 Atualização Adaptativa da Grade Durante o Treinamento	11
2.4 Transformada B-splines Direta como Filtro de Suavização	11
3 Trabalhos Relacionados	15
3.1 Pesquisa Bibliográfica	15
3.1.1 Fundamentos do Traçamento de Raios Sísmicos	15
3.1.2 Métodos Numéricos Tradicionais	15
3.1.3 Fundamentos de Redes Neurais e PINNs	16
3.1.4 Fundamentos das Redes de Kolmogorov-Arnold (KANs)	16
3.2 Estado da Arte	16
3.2.1 Evolução dos Métodos Numéricos em Geofísica	16
3.2.2 Aprendizado de Máquina Aplicado ao Traçamento de Raios	17
3.2.3 Aplicações Recentes de KANs em Física	17
3.3 Contribuições do Trabalho Proposto	17
4 Metodologia	19
4.1 Parte 1: Obtenção do Conjunto de Dados Sintéticos	19
4.1.1 Aplicação do Filtro B-Splines	21
4.1.2 Traçamento de Raios Sísmicos	22
4.1.3 Criação do Regularizador	25

4.1.4	Divisão do Conjunto de Dados	27
4.2	Parte 2: Treinamento e Validação do Modelo	29
4.2.1	Formulação da Função de Perda	29
4.2.2	Arquitetura e Implementação	31
4.2.3	Processo de Treinamento	32
4.2.4	Ajuste do Tamanho da Grade na Rede de Kolmogorov-Arnold	34
4.2.5	Ajuste do Fator de Subamostragem no Pré-processamento do Campo de Velocidade	36
4.2.6	Validação do Modelo	36
4.2.7	Outras Estratégias de Regularização	37
4.2.8	Considerações sobre Reprodutibilidade	38
4.3	Parte 3: Avaliação do Modelo	39
4.3.1	Métricas de Avaliação Quantitativa	39
4.3.2	Implementação das Métricas de Avaliação	41
4.3.3	Justificativa da Escolha das Métricas	41
4.3.4	Avaliação Qualitativa	42
4.3.5	Protocolo de Avaliação	42
4.3.6	Considerações sobre Avaliação em Modelos Híbridos PIKAN	43
5	Experimentos e Resultados	44
5.1	Modelo Parametrizado	44
5.1.1	Geração dos Dados Sintéticos	45
5.1.2	Geração do Conjunto de Dados Completo	47
5.1.3	Visualização dos Dados Gerados	48
5.1.4	Refinamento de Hiperparâmetro: ajuste do tamanho da grade na KAN	49
5.1.5	Treinamento e Avaliação do Melhor Modelo	53
5.2	Modelo Não Parametrizado	56
5.2.1	Geração dos Dados Sintéticos: Marmousi	57
5.2.2	Geração e Visualização do Conjunto de Dados para o Modelo do Marmousi	58
5.2.3	Treinamento e Avaliação do Modelo	59
5.2.4	Comparação entre PIKAN e PINN tradicional	66
6	Conclusão	70
6.1	Síntese das contribuições	70
6.2	Limitações do trabalho	71
6.3	Trabalhos futuros	71
6.4	Considerações finais	72
	Referências bibliográficas	73

Capítulo 1

Introdução

A modelagem de propagação de ondas sísmicas é um componente fundamental para o imageamento sísmico e a caracterização de reservatórios na indústria de petróleo e gás (Sheriff 2002, Yilmaz 2001). Dentre as diversas técnicas de modelagem, o traçamento de raios sísmicos destaca-se como uma abordagem eficiente e amplamente utilizada para simular a propagação de ondas em meios heterogêneos (Muniz 2016). Esta técnica consiste na resolução de um sistema de equações diferenciais não-lineares que aproximam a Lei de Snell para estimar as trajetórias dos raios e os tempos de trânsito em modelos de velocidade heterogêneos e anisotrópicos (Červený 2001, Virieux et al. 1988, Červený 1989).

O traçamento de raios sísmicos é particularmente relevante em contextos onde a precisão na determinação das trajetórias de propagação das ondas é crucial, como na migração pré-empilhamento em profundidade (PSDM, do inglês *Post Stack Depth Migration*), na tomografia sísmica e na análise de atributos sísmicos (Wang 2013, Romero et al. 2024). Tradicionalmente, este problema é abordado por meio de métodos numéricos convencionais, como o método de Runge-Kutta, que resolvem as equações canônicas do sistema Hamiltoniano que descrevem a trajetória dos raios no meio (Margrave 2003, Margrave & Lamoureux 2019). Embora estes métodos forneçam soluções precisas em condições ideais, eles apresentam limitações significativas quando aplicados a modelos de velocidade com alto contraste ou variabilidade espacial acentuada, situações comuns em ambientes geológicos complexos (Koketsu & Sekine 1998, Carcione et al. 2002).

Nestas condições desafiadoras, os métodos numéricos tradicionais frequentemente perdem estabilidade ou demandam esforço computacional excessivo, comprometendo sua aplicabilidade prática em cenários geofísicos realistas (Vesnaver 1996, Rawlinson & Sambridge 2004). Além disso, a necessidade de reiniciar o processo de integração para cada condição inicial nova (como posição da fonte ou ângulo de disparo) torna estes métodos computacionalmente onerosos quando múltiplos raios precisam ser calculados, como é o caso em aplicações de imageamento sísmico (Virieux & Operto 2009).

Diante destas limitações, nos últimos anos, tem havido um crescente interesse no desenvolvimento de abordagens alternativas baseadas em aprendizado de máquina para a solução de problemas de traçamento de raios sísmicos (Zhu et al. 2019, Smith et al. 2021, Duarte et al. 2023). Em particular, as Redes Neurais com Informação Física (PINNs, do inglês *Physics-Informed Neural Networks*) emergiram como uma linha de pesquisa promissora (Raissi et al. 2019, Karniadakis et al. 2021). Estas redes incorporam as equações

físicas que governam o fenômeno estudado diretamente na função de perda, durante o treinamento, permitindo que o modelo aprenda soluções fisicamente consistentes mesmo com acesso limitado a dados rotulados (Cuomo et al. 2022, Moseley et al. 2020a).

Apesar dos avanços proporcionados pelas PINNs, estas ainda enfrentam desafios significativos quando aplicadas ao traçamento de raios sísmicos (bin Waheed et al. 2021, Moseley et al. 2020b). Problemas de convergência e generalização são recorrentes, especialmente quando se tenta incorporar condições iniciais variáveis como entradas do modelo (Wang et al. 2022). Consequentemente, as implementações atuais de PINNs para traçamento de raios frequentemente limitam-se a aprender uma única trajetória com condições iniciais fixas, restringindo severamente sua generalização e escalabilidade para aplicações práticas em geofísica (Smith et al. 2021, Duarte et al. 2023, bin Waheed et al. 2021).

O presente trabalho propõe uma nova abordagem para superar estas limitações, empregando Redes de Kolmogorov-Arnold com Informação Física (PIKANs, do inglês *Physics-Informed Kolmogorov-Arnold Networks*) para o traçamento de raios sísmicos. As Redes de Kolmogorov-Arnold (KANs, do inglês *Kolmogorov-Arnold Networks*) representam uma arquitetura neural recentemente desenvolvida, inspirada no teorema da representação de Kolmogorov-Arnold, que afirma que qualquer função contínua multivariada pode ser exatamente representada como uma superposição finita de funções contínuas univariadas e a operação de adição (Kolmogorov 1957, Liu et al. 2025).

Diferentemente das redes neurais convencionais, onde as arestas da rede carregam pesos escalares e os nós aplicam funções de ativação fixas e não-lineares, as KANs atribuem funções univariadas treináveis às arestas, enquanto os nós realizam simples somas (Liu et al. 2025, Yamak et al. 2025). Esta estrutura alinha-se com a representação de Kolmogorov-Arnold, enfatizando composições e somas de funções univariadas para resolver funções multivariadas de maior complexidade. Tal arquitetura oferece vantagens significativas em termos de capacidade de interpolação, interpretabilidade e eficiência de treinamento (Liu et al. 2025, Hadj Kilani 2024).

1.1 Hipótese Científica e Objetivo Geral da Dissertação

A hipótese central deste trabalho é que a integração das KANs com princípios de informação física pode superar as limitações das abordagens existentes para o traçamento de raios sísmicos, permitindo a modelagem eficiente e precisa de múltiplas trajetórias de raios, de diferentes condições iniciais, em modelos de velocidade complexos. Especificamente, propõe-se que as PIKANs, ao combinarem a capacidade de representação superior das KANs com a incorporação de restrições físicas durante o treinamento, podem generalizar efetivamente para condições iniciais não vistas durante o treinamento, mantendo consistência física e precisão numérica.

O objetivo geral do trabalho é desenvolver e validar um *framework* baseado em PIKANs para estimar trajetórias de raios sísmicos a partir de condições iniciais variáveis. O modelo é treinado com uma função de perda composta, que combina erro orientado por dados, penalizações físicas via diferenciação automática e um esquema de ponderação adaptativa. A abordagem visa aplicações em imageamento sísmico, inversão tomográfica e monitoramento de reservatórios, contribuindo para soluções mais robustas e eficientes

em geofísica computacional.

1.2 Objetivos Específicos

Com base na hipótese científica e no objetivo geral estabelecidos, definem-se os seguintes objetivos específicos deste trabalho:

- Desenvolver um *framework* baseado em Redes de Kolmogorov-Arnold com Informação Física para o traçamento de raios sísmicos em meios heterogêneos e complexos;
- Implementar uma função de perda composta que integre erro orientado por dados, penalizações físicas por meio de diferenciação automática e um esquema de ponderação adaptativa conforme a densidade de amostragem local;
- Avaliar a capacidade de generalização do modelo para condições iniciais não vistas durante o treinamento, como diferentes posições de fonte e ângulos de disparo;
- Integrar uma abordagem de reconstrução de modelos de velocidade utilizando interpolação B-splines a partir de dados paramétricos esparsamente amostrados;
- Validar a eficácia do *framework* proposto em cenários sintéticos, incluindo modelos parametrizados e o modelo Marmousi, considerando métricas de precisão, eficiência computacional e aderência física;

1.3 Estrutura do texto

O restante deste texto está organizado da seguinte forma:

- O Capítulo 2 desenvolve o embasamento teórico necessário, abordando os princípios do traçamento de raios sísmicos, formulação Hamiltoniana, Redes Neurais com Informação Física e a teoria por trás das Redes de Kolmogorov-Arnold;
- O Capítulo 3 oferece uma revisão bibliográfica sobre o traçamento de raios sísmicos, métodos numéricos clássicos e abordagens baseadas em aprendizado de máquina, além de realizar uma análise do estado da arte;
- O Capítulo 4 detalha a metodologia proposta, incluindo a arquitetura PIKAN, o esquema de ponderação adaptativo e a técnica de reconstrução do modelo de velocidade via interpolação B-splines, bem como descreve os experimentos realizados para validar a abordagem, incluindo o desenho experimental, os conjuntos de dados utilizados, as métricas de avaliação e as comparações com métodos numéricos tradicionais;
- O Capítulo 5 apresenta e discute os resultados obtidos, avaliando o desempenho do método em termos de precisão, eficiência computacional e capacidade de generalização, além de analisar casos específicos e limitações observadas;
- O Capítulo 6 conclui o trabalho, resumindo as contribuições, discutindo limitações e apontando direções para pesquisas futuras, incluindo possíveis extensões e aplicações adicionais da abordagem proposta.

Capítulo 2

Fundamentação Teórica

Este capítulo apresenta as ideias fundamentais que embasam a metodologia proposta, incluindo o traçamento de raios sísmicos, as Redes Neurais com Informação Física e as Redes de Kolmogorov-Arnold. Estes conceitos servem como base teórica para o desenvolvimento da metodologia proposta neste trabalho, detalhando os princípios matemáticos e computacionais que sustentam cada componente.

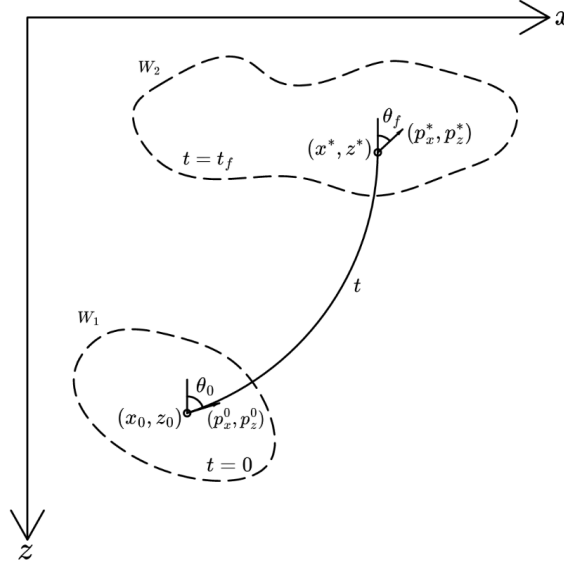
2.1 Traçamento de Raios Sísmicos

O traçamento de raios sísmicos é uma técnica fundamental em geofísica para modelar a propagação de ondas através de meios heterogêneos. Dado que o objetivo deste trabalho é prever trajetórias de raios prospectivas (*forward raypaths*), é utilizada a formulação Hamiltoniana — método bem estabelecido em estudos de traçamento de raios sísmicos (Virieux et al. 1988, Červený 1989, Billette & Lambaré 1998). Esta abordagem descreve a trajetória do raio como a evolução de um sistema no espaço de fase $(\mathbf{x}, \mathbf{p})$, onde $\mathbf{x}$ é a posição espacial e $\mathbf{p}$ é o vetor de vagarosidade — cuja magnitude é o recíproco do módulo de velocidade.

A Figura 2.1 ilustra o conceito fundamental do traçamento de raios sísmicos no espaço bidimensional x - z . O trajeto do raio é representado como uma curva contínua que evolui entre duas regiões, W_1 e W_2 , com velocidades diferentes. No instante inicial $t = 0$, o raio parte do ponto (x_0, z_0) , com vetor de vagarosidade $\mathbf{p}^0 = (p_x^0, p_z^0)$ e ângulo inicial θ_0 em relação à vertical, inserido na região W_1 . O raio se propaga ao longo do tempo t segundo a formulação Hamiltoniana, que descreve sua evolução no espaço de fase $(\mathbf{x}, \mathbf{p})$. No instante final $t = t_f$, o raio alcança o ponto (x^*, z^*) , dentro da região W_2 , com vetor de vagarosidade $\mathbf{p}^* = (p_x^*, p_z^*)$ e ângulo final θ_f diferentes dos valores iniciais, indicando a mudança de velocidade do meio. Esta representação gráfica evidencia como a trajetória do raio sísmico é determinada pela evolução simultânea da posição e da vagarosidade, fundamentando a abordagem adotada neste trabalho para a previsão de trajetórias prospectivas de raios sísmicos.

A função Hamiltoniana H é introduzida como uma invariante ao longo da trajetória do raio em um meio isotrópico, isto é, seu valor permanece constante durante a propagação. Essa invariância garante que a relação de dispersão local da onda seja satisfeita ao longo de todo o percurso do raio e está diretamente associada à formulação da equação eikonal.

Figura 2.1: Trajetória de um raio sísmico no plano $x-z$, do ponto inicial (x_0, z_0) na região W_1 ao ponto final (x^*, z^*) na região W_2 , com ângulos inicial e final θ_0 e θ_f , respectivamente.



Fonte: Elaborado pelo autor (2025).

O Hamiltoniano é definido como (Lambaré et al. 1996):

$$H(\mathbf{x}, \mathbf{p}) = \frac{1}{2} (\|\mathbf{p}\|^2 v^2(\mathbf{x}) - 1), \quad (2.1)$$

onde $\mathbf{x} = (x, z)$ denota a posição espacial, $v(\mathbf{x})$ é a magnitude da velocidade de propagação da onda no ponto $\mathbf{x}$, e $\mathbf{p} = (p_x, p_z)$ corresponde ao vetor de vagarosidade (*slowness*), que é normal à frente de onda e cuja magnitude $\|\mathbf{p}\|$ é a vagarosidade local $1/v(\mathbf{x})$. A condição $H = 0$ ao longo da trajetória do raio é equivalente à equação eikonal (Červený 2001).

O vetor de vagarosidade $\mathbf{p}$ está relacionado ao gradiente do tempo de trânsito $t(\mathbf{x})$ pela equação $\mathbf{p} = \nabla t(\mathbf{x})$. A evolução da trajetória do raio no espaço de fase é governada pelas equações canônicas de Hamilton, que descrevem como a posição $\mathbf{x}$ e o vetor de vagarosidade $\mathbf{p}$ variam ao longo de um parâmetro τ (frequentemente, o próprio tempo de trânsito t):

$$\frac{d\mathbf{x}}{d\tau} = \nabla_{\mathbf{p}} H = \frac{\partial H}{\partial \mathbf{p}} = v^2(\mathbf{x}) \mathbf{p}, \quad (2.2)$$

$$\frac{d\mathbf{p}}{d\tau} = -\nabla_{\mathbf{x}} H = -\frac{\partial H}{\partial \mathbf{x}} = -\|\mathbf{p}\|^2 v(\mathbf{x}) \nabla v(\mathbf{x}). \quad (2.3)$$

Adotando o tempo de trânsito t como o parâmetro de integração (i.e., $d\tau = dt$), e utilizando a relação eikonal $\|\mathbf{p}\|^2 = 1/v^2(\mathbf{x})$, as equações canônicas podem ser reescritas na forma comumente utilizada:

$$\frac{d\mathbf{x}}{dt} = v^2(\mathbf{x})\mathbf{p}, \quad (2.4)$$

$$\frac{d\mathbf{p}}{dt} = -\frac{1}{v(\mathbf{x})}\nabla v(\mathbf{x}). \quad (2.5)$$

As equações 2.4 e 2.5 formam um sistema de equações diferenciais ordinárias (EDOs) de primeira ordem definido no espaço de fases de dimensão $2d$, onde d representa a dimensão do espaço físico, tipicamente 2 ou 3. Assim, no caso bidimensional, a dinâmica do raio ocorre em um espaço de fases de quatro dimensões, descrito pelas variáveis (x, z, p_x, p_z) . Esse sistema pode ser resolvido numericamente por métodos de integração padrão, como Runge-Kutta ou métodos adaptativos como o Dormand-Prince. Esses métodos são eficientes para traçar a evolução da trajetória do raio $(\mathbf{x}(t), \mathbf{p}(t))$, a partir de uma condição inicial $(\mathbf{x}_0, \mathbf{p}_0)$ por execução, permitindo a obtenção do tempo de trânsito e a geometria do raio.

2.2 Redes Neurais com Informação Física (PINNs)

As Redes Neurais com Informação Física (PINNs) (Raissi et al. 2019) representam uma classe de modelos de aprendizado profundo que integram o conhecimento de leis físicas, codificadas como equações diferenciais, ao processo de aprendizado baseado em dados. Diferentemente das redes neurais convencionais, que são ajustadas exclusivamente aos dados, as PINNs utilizam as equações governantes como um termo de regularização na função de perda, promovendo soluções que são simultaneamente consistentes com as observações e com os princípios físicos subjacentes.

Considere um sistema físico descrito por uma Equação Diferencial Parcial (EDP) não linear geral:

$$\mathcal{N}[u](\mathbf{z}) = 0, \quad \mathbf{z} \in \Omega, \quad (2.6)$$

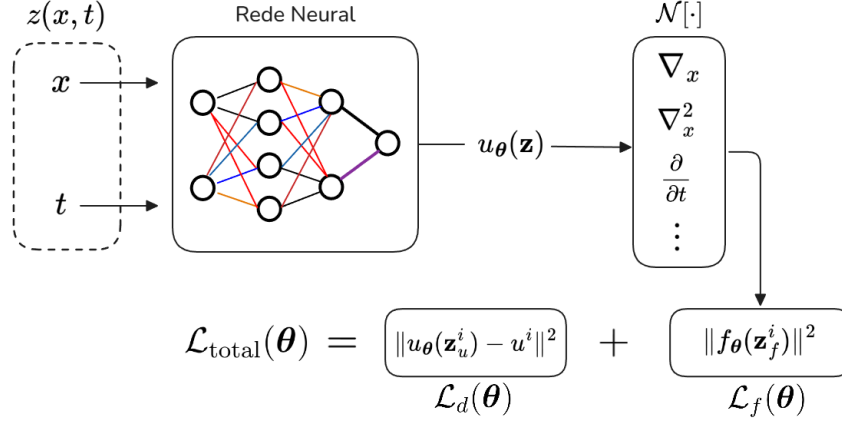
sujeita a condições de contorno $\mathcal{B}[u](\mathbf{z}) = g(\mathbf{z})$ em $\partial\Omega$. Aqui, $u(\mathbf{z})$ é a solução desconhecida (e.g., campo de pressão, deslocamento, temperatura), $\mathbf{z} = (\mathbf{x}, t)$ representa as coordenadas espaço-temporais, $\mathcal{N}[\cdot]$ é um operador diferencial (possivelmente não linear), Ω é o domínio da EDP, e $\mathcal{B}[\cdot]$ é um operador de condição de contorno (e.g., Dirichlet, Neumann, Robin).

Na metodologia PINN, a solução $u(\mathbf{z})$ é aproximada por uma rede neural profunda $u_{\boldsymbol{\theta}}(\mathbf{z})$, onde $\boldsymbol{\theta}$ denota o conjunto de parâmetros treináveis (pesos e vieses) da rede. A capacidade da rede neural de aproximar funções de maior complexidade permite representar soluções u com geometrias e variações de maior complexidade. A chave da abordagem PINN consiste na definição de uma função de perda composta que penaliza tanto o desvio dos dados observados quanto a violação das leis físicas.

O resíduo da EDP, avaliado pela aproximação da rede neural, é definido como:

$$f_{\boldsymbol{\theta}}(\mathbf{z}) := \mathcal{N}[u_{\boldsymbol{\theta}}](\mathbf{z}). \quad (2.7)$$

Figura 2.2: Diagrama representando o funcionamento das Redes Neurais com Informação Física (PINNs).



Fonte: Elaborado pelo autor (2025).

As derivadas parciais de u_{θ} em relação a $\mathbf{z}$, necessárias para o cálculo do operador $\mathcal{N}$, são obtidas de maneira eficiente por meio de diferenciação automática (AD, do inglês *Automatic Differentiation*). A AD permite o cálculo exato (com exceção de erros devido à precisão numérica) das derivadas das funções computadas, sendo uma ferramenta indispensável nos *frameworks* modernos de aprendizado profundo (Baydin et al. 2018).

A função de perda $\mathcal{L}_{\text{total}}(\theta)$ a ser minimizada durante o treinamento é tipicamente uma soma ponderada de múltiplos termos. A Figura 2.2 representa, de maneira simplificada, como obter essa função. Define-se $\mathcal{L}_{\text{total}}(\theta)$ como

$$\mathcal{L}_{\text{total}}(\theta) = \lambda_d \mathcal{L}_d(\theta) + \lambda_f \mathcal{L}_f(\theta) + \lambda_c \mathcal{L}_c(\theta), \quad (2.8)$$

onde:

- $\mathcal{L}_d(\theta)$ mede a discrepância entre as previsões da rede u_{θ} e os N_u dados de treinamento disponíveis $\{(\mathbf{z}_u^i, u^i)\}_{i=1}^{N_u}$ (e.g., medições experimentais, condições iniciais):

$$\mathcal{L}_d(\theta) = \frac{1}{N_u} \sum_{i=1}^{N_u} \|u_{\theta}(\mathbf{z}_u^i) - u^i\|^2. \quad (2.9)$$

Este termo ancora a solução aos dados observados.

- $\mathcal{L}_f(\theta)$ penaliza o resíduo da EDP em um conjunto de N_f pontos de avaliação $\{\mathbf{z}_f^i\}_{i=1}^{N_f}$ amostrados dentro do domínio Ω :

$$\mathcal{L}_f(\theta) = \frac{1}{N_f} \sum_{i=1}^{N_f} \|f_{\theta}(\mathbf{z}_f^i)\|^2. \quad (2.10)$$

Este termo impõe que a solução da rede neural satisfaça (aproximadamente) a equação física governante em todo o domínio.

- $\mathcal{L}_c(\boldsymbol{\theta})$ impõe as condições de contorno (e/ou iniciais) em N_b pontos $\{\mathbf{z}_b^i\}_{i=1}^{N_b}$ na fronteira $\partial\Omega$:

$$\mathcal{L}_c(\boldsymbol{\theta}) = \frac{1}{N_b} \sum_{i=1}^{N_b} \|\mathcal{B}[u_{\boldsymbol{\theta}}](\mathbf{z}_b^i) - g(\mathbf{z}_b^i)\|^2. \quad (2.11)$$

Os pesos λ_d , λ_f , λ_c são hiperparâmetros que ajustam a importância relativa de cada termo na perda total. A escolha adequada desses pesos é crucial e pode ser desafiadora, pois um desequilíbrio pode levar a uma convergência pobre ou a soluções que não respeitam adequadamente a física ou os dados. Estratégias de ponderação adaptativa, como as baseadas na magnitude dos gradientes de cada termo da função de perda (Wang et al. 2021) ou em abordagens inspiradas em otimização multiobjetivo (Eivazi et al. 2022), buscam ajustar dinamicamente esses pesos durante o treinamento para melhorar a convergência e a precisão.

A eficácia das PINNs depende criticamente da capacidade do otimizador (e.g., SGD, Adam, L-BFGS) de navegar na função de perda $\mathcal{L}_{\text{total}}(\boldsymbol{\theta})$ e encontrar um bom mínimo local. Esta função pode ser altamente não convexa e apresentar múltiplos mínimos locais, platôs e gradientes evanescentes ou explosivos, especialmente para EDPs de maior complexidade ou não lineares (Krishnapriyan et al. 2021). Técnicas de regularização adicionais, como L1 e/ou L2 sobre os pesos $\boldsymbol{\theta}$, inicialização cuidadosa dos pesos, estratégias de amostragem adaptativa dos pontos de avaliação e a escolha da arquitetura da rede (profundidade, largura, funções de ativação) são aspectos importantes para mitigar esses desafios e melhorar o desempenho e a robustez das PINNs (Cuomo et al. 2022).

A incorporação da física como um regularizador implícito permite que as PINNs aprendam soluções fisicamente plausíveis mesmo com dados esparsos, ruidosos ou ausentes em certas regiões do domínio, tornando-as particularmente promissoras para problemas inversos (e.g., estimativa de parâmetros da EDP a partir de dados), interpolação e integração de dados, e cenários onde a obtenção de dados experimentais ou simulações numéricas de alta fidelidade é custosa (Karniadakis et al. 2021).

2.3 Redes de Kolmogorov-Arnold (KAN)

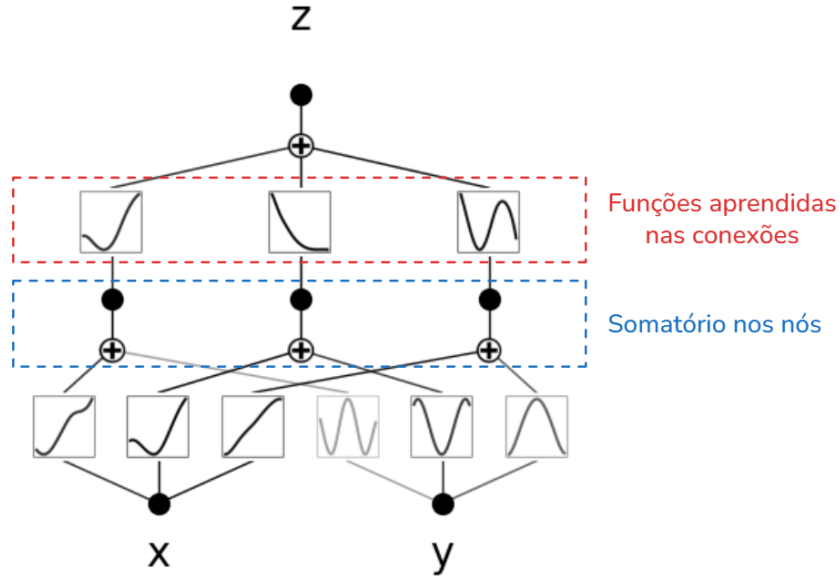
As Redes de Kolmogorov-Arnold (KANs) constituem uma arquitetura de rede neural recentemente proposta por (Liu et al. 2025), que se diferencia fundamentalmente dos Perceptrons Multicamadas (MLPs) tradicionais ao basear sua estrutura no teorema da representação de Kolmogorov-Arnold (Kolmogorov 1957). Este teorema afirma que qualquer função contínua multivariada $f : [0, 1]^n \rightarrow \mathbb{R}$ pode ser representada exatamente como uma superposição finita de funções contínuas univariadas e a operação de adição. Mais precisamente, existem $2n + 1$ funções externas contínuas $\Phi_q : \mathbb{R} \rightarrow \mathbb{R}$ e $(2n + 1) \times n$ funções internas contínuas $\phi_{q,p} : [0, 1] \rightarrow \mathbb{R}$ tais que:

$$f(x_1, \dots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^n \phi_{q,p}(x_p) \right). \quad (2.12)$$

Inspiradas por esta representação, as KANs propõem uma arquitetura onde as funções

de ativação não lineares são localizadas nas conexões (arestas) entre os neurônios, como ilustrado na Figura 2.3, em vez de nos próprios neurônios (nós), como nas MLPs. Além disso, essas funções de ativação nas arestas são parametrizadas e treináveis, tipicamente utilizando splines.

Figura 2.3: Arquitetura simplificada da KAN, onde as entradas são as variáveis x e y , e a saída z está relacionada à estimativa da função desejada, conforme a estrutura da rede.



Fonte: Adaptado de Liu et al. (2024).

Considere uma KAN com L camadas, onde a camada l possui n_l neurônios. A saída $x_{l+1,j}$ do j -ésimo neurônio na camada $l+1$ é calculada como a soma das saídas das funções de ativação $\phi_{l,j,i}$ aplicadas às saídas $x_{l,i}$ dos neurônios da camada anterior:

$$x_{l+1,j} = \sum_{i=1}^{n_l} \phi_{l,j,i}(x_{l,i}), \quad \text{para } j = 1, \dots, n_{l+1}. \quad (2.13)$$

Cada função $\phi_{l,j,i} : \mathbb{R} \rightarrow \mathbb{R}$ é uma função univariada treinável. Uma escolha comum para parametrizar essas funções é através de B-splines:

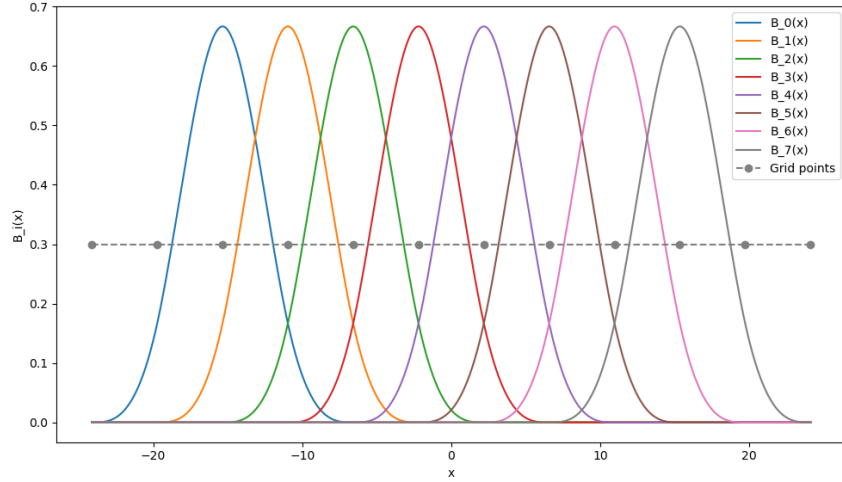
$$\phi(x) = \lambda_{\text{base}} \cdot \text{base}(x) + \lambda_{\text{spline}} \cdot \text{spline}(x; \mathbf{t}, k) \quad (2.14)$$

onde $\text{spline}(x; \mathbf{t}, k)$ é a expansão em B-splines de grau k com G intervalos, também chamados de tamanho da grade, definidos pelo vetor de nós $\mathbf{t}$, tal que,

$$\text{spline}(x; \mathbf{t}, k) = \sum_{i=1}^{G+d} c_i B_i(x; \mathbf{t}, k), \quad (2.15)$$

c_i são os coeficientes treináveis da spline, B_i são as funções de base B-splines, como na Figura 2.4, que serão apresentadas, com mais detalhes, na Seção 2.4. E $\text{base}(x)$ é uma

Figura 2.4: Exemplo das funções de base B-splines (B_k) após a expansão da grade para $G = 5$ e com splines de ordem $k = 3$.



Fonte: Adaptado de Liu et al. (2024).

função base (e.g., linear ou sigmoideal) com peso λ_{base} . Os pesos λ_{base} e λ_{spline} também são treináveis. A utilização de B-splines permite que as funções de ativação aprendam formas de maior complexidade e se adaptem localmente aos dados, com a resolução controlada pelo número de nós e pelo grau da spline (de Boor 1978).

Em notação matricial, a transformação da camada l para a camada $l + 1$ pode ser vista como uma aplicação de uma “matriz funcional” Φ_l ao vetor de saída $\mathbf{x}_l$:

$$\mathbf{x}_{l+1} = \Phi_l(\mathbf{x}_l). \quad (2.16)$$

A saída final da rede $\mathbf{x}_L$ após L camadas é a composição dessas transformações funcionais:

$$\mathbf{x}_L = (\Phi_{L-1} \circ \Phi_{L-2} \circ \dots \circ \Phi_0)(\mathbf{x}_0). \quad (2.17)$$

Esta estrutura difere significativamente da composição de transformações lineares e ativações não lineares fixas dos MLPs, definida pela Equação 2.18.

$$\mathbf{x}_L = (\sigma_{L-1} \circ P_{L-1} \circ \sigma_{L-2} \circ P_{L-2} \circ \dots \circ \sigma_0 \circ P_0)(\mathbf{x}_0), \quad (2.18)$$

onde σ_i e P_i são a função de ativação e a matriz de pesos da camada i da rede MLP, respectivamente.

As KANs oferecem vantagens potenciais em termos de precisão e interpretabilidade. Teoricamente, KANs podem alcançar melhor escalabilidade em relação à precisão do que MLPs para certas classes de funções, pois a complexidade (número de parâmetros) cresce mais lentamente com a dimensão da entrada n (Liu et al. 2025). A interpretabilidade advém da possibilidade de visualizar as funções univariadas $\phi_{l,j,i}$ aprendidas, o que pode revelar como a rede decompõe a função multivariada e quais interações entre variáveis

são importantes. Além disso, a arquitetura permite a adaptação dinâmica da resolução das splines, o que pode aprimorar a precisão da aproximação em regiões específicas do espaço de entrada.

Contudo, o treinamento das KANs envolve otimizar um número maior de parâmetros (coeficientes das splines) e pode exigir mais recursos computacionais em comparação ao treinamento de MLPs equivalentes em relação ao número de camadas e neurônios. A estabilidade numérica e a convergência durante o treinamento também são áreas de pesquisa ativa (Dhiman 2024).

2.3.1 Atualização Adaptativa da Grade Durante o Treinamento

Um mecanismo importante na implementação da KAN é o de atualização adaptativa da grade durante o treinamento, conforme proposto por (Liu et al. 2025). Este mecanismo ajusta a distribuição dos nós da grade para concentrar mais recursos em regiões do domínio com maior complexidade funcional.

Formalmente, para cada função unidimensional g , a posição dos nós é atualizada periodicamente de acordo com a distribuição empírica dos dados de entrada e a magnitude dos gradientes da função de perda em relação aos coeficientes B-splines:

$$t_i^{\text{nov}} = Q_\rho((i-1)/(G-1)), \quad i = 1, 2, \dots, G \quad (2.19)$$

onde Q_ρ é a função quantil da distribuição de densidade ρ , definida como:

$$\rho(x) \propto \sum_{j: x_j \in \mathcal{H}(x)} \left| \frac{\partial \mathcal{L}}{\partial g(x_j)} \frac{\partial g(x_j)}{\partial x} \right| \quad (2.20)$$

onde $\mathcal{H}(x)$ é uma vizinhança de x , e a soma é calculada sobre todas as amostras de entrada que caem nesta vizinhança.

2.4 Transformada B-splines Direta como Filtro de Suavização

As B-splines são funções polinomiais por partes que constituem uma base popular para a representação e aproximação de funções, devido às suas propriedades vantajosas: suporte local compacto, suavidade controlável pelo grau n , partição da unidade e estabilidade numérica (de Boor 1978). A transformada B-splines direta é um algoritmo computacionalmente eficiente para determinar os coeficientes $c[k]$ de uma representação B-splines $f(x) = \sum_k c[k] \beta^n(x-k)$, que interpola ou aproxima um sinal discreto $s[k]$ amostrado em pontos inteiros, onde $\beta^n(x)$ é a função B-splines causal de grau n centrada na origem.

A interpolação exata ($f(k) = s[k]$) requer a solução de um sistema linear envolvendo a matriz de convolução com as amostras da B-splines $\beta^n[k]$. Essa deconvolução pode ser realizada de forma eficiente como um filtro de Resposta ao Impulso Infinito (IIR, do inglês *Infinite Impulse Response*), explorando a fatorização da transformada z da sequência

$\beta^n[k]$ (Unser et al. 1993b, Unser et al. 1993a).

A formulação apresentada no texto original, utilizando as recursões causal e anticausal com o polo $z_1 = -2 + \sqrt{3}$, corresponde a uma aplicação específica dessa teoria, resultando em um filtro de suavização. Este filtro aproxima a projeção do sinal $s[k]$ no espaço das splines cúbicas ($n = 3$).

A função de base B-splines cúbica $\beta^3(x)$ é definida como:

$$\beta^3(x) = \frac{1}{6} \begin{cases} (2 - |x|)^3 & 1 \leq |x| < 2 \\ 4 - 6x^2 + 3|x|^3 & 0 \leq |x| < 1 \\ 0 & |x| \geq 2 \end{cases} \quad (2.21)$$

O filtro IIR, correspondente à interpolação exata, tem polos estáveis e instáveis. A aplicação sequencial de filtros recursivos baseados apenas nos polos estáveis (como z_1) de forma causal e anticausal implementa um filtro de suavização simétrico e de fase zero, cujo efeito é aproximar o sinal original pela sua projeção no espaço das splines (Unser 1999).

As equações para as recursões causal (c^+) e anticausal (c^-) são apresentadas a seguir:

- **Recursão Causal:** A recursão causal é expressa pela seguinte equação:

$$c^+[k] = \alpha s[k] + z_1 \cdot c^+[k-1], \quad \text{para } k = 1, 2, \dots, N-1. \quad (2.22)$$

A condição inicial $c^+[0]$ deve ser devidamente inicializada para garantir o tratamento adequado das condições de contorno. Uma abordagem possível é utilizar a soma da série geométrica para simular um sinal infinito ou aplicar condições de contorno específicas.

- **Recursão Anticausal:** A recursão anticausal é dada por:

$$c^-[k] = z_1 \cdot (c^-[k+1] - c^+[k]), \quad \text{para } k = N-2, N-3, \dots, 0. \quad (2.23)$$

A condição inicial $c^-[N-1]$ também requer tratamento especial nas bordas.

- **Resultado Final:** O resultado final é obtido a partir de $c[k] = c^-[k]$, fornecendo os coeficientes B-splines que representam uma versão suavizada do sinal original $s[k]$. A função contínua suavizada é, então, dada por:

$$f_{\text{smooth}}(x) = \sum_k c[k] \beta^3(x-k). \quad (2.24)$$

No entanto, a aplicação direta do método de interpolação descrito em dados discretos com bordas pode introduzir artefatos e descontinuidades. Para mitigar esses problemas, o espelhamento simétrico (do inglês, *mirror padding*) é empregado nas bordas da matriz de coeficientes do modelo de velocidade antes da interpolação. Este procedimento garante a continuidade do modelo de velocidades após a reconstrução, resultando em um comportamento adequado das derivadas nas regiões da borda do campo de velocidade.

O espelhamento simétrico consiste em estender a matriz de coeficientes, replicando valores internos de forma simétrica nas bordas, conforme o exemplo ilustrado na Figura 2.5. Esta técnica é matematicamente justificada pela necessidade de estender o

Figura 2.5: Exemplo da aplicação de um *mirror padding* em um conjunto de dados bidimensional, como o campo de velocidade. Em vermelho estão os valores do conjunto de dados original, enquanto nos azuis estão os dados espelhados.

11	10	9	10	11	12	11	10
7	6	5	6	7	8	7	6
3	2	1	2	3	4	3	2
7	6	5	6	7	8	7	6
11	10	9	10	11	12	11	10
15	14	13	14	15	16	15	14
11	10	9	10	11	12	11	10
7	6	5	6	7	8	7	6

Fonte: Elaborado pelo autor (2025).

suporte local das funções base $\beta^3(x)$ das B-splines, que têm um suporte finito definido no intervalo $[-2, 2]$ para o caso cúbico ($n = 3$). Essas funções de base precisam de dados além dos limites da matriz original para garantir uma interpolação suave nas bordas, sem introduzir descontinuidade ou erros numéricos. A implementação do espelhamento simétrico para um *mirror padding* de 1 é dada por

$$c^{\text{espelhado}}(i, j) = c(n(i), m(j)),$$

$$n(i) = \begin{cases} 2, & i = 1, \\ i - 1, & 2 \leq i \leq N + 1, \\ N, & i = N + 2, \end{cases} \quad m(j) = \begin{cases} 2, & j = 1, \\ j - 1, & 2 \leq j \leq M + 1, \\ M, & j = M + 2. \end{cases} \quad (2.25)$$

onde $c^{\text{espelhado}}(i, j)$ corresponde ao elemento da linha i , coluna j da matriz de coeficientes de velocidade estendida pelo espelhamento simétrico. Os índices $i \in \{1, 2, \dots, N + 2\}$ e $j \in \{1, 2, \dots, M + 2\}$ percorrem a nova matriz $c^{\text{espelhado}}$ com dimensões $(N + 2) \times (M + 2)$, sendo $N \times M$ as dimensões da matriz original c . Por fim, $n(i)$ e $m(j)$ são funções que mapeiam, respectivamente, o índice da linha e da coluna da matriz original c a partir de um índice da matriz espelhada $c^{\text{espelhado}}$.

A ausência de tais valores resultaria em uma interpolação inconsistente e na introdução de descontinuidades artificiais nas bordas. Ao garantir a suavização e a continuidade das derivadas, o espelhamento simétrico assegura que o campo de velocidade interpolado

$(v(\mathbf{x}))$ e suas derivadas parciais (como $\frac{\partial v}{\partial x}$ e $\frac{\partial v}{\partial z}$) se comportem de maneira previsível e suave em toda a extensão do domínio, incluindo as bordas. Isso é particularmente relevante para o cálculo do gradiente do campo de velocidade (∇v) utilizado na integração das equações de traçamento de raios (Unser et al. 1993a, Unser 1999, de Boor 1978).

A transformada B-splines atua como um filtro passa-baixas eficaz. A resposta em frequência do filtro de suavização B-splines de grau n aproxima uma função sinc elevada à potência $n + 1$, proporcionando uma atenuação suave das altas frequências (Unser 1999). Essa propriedade é extremamente útil para a regularização de dados ruidosos e para garantir a diferenciabilidade de campos (como modelos de velocidade sísmica) representados a partir de dados discretos. A representação B-splines garante que o campo reconstruído $f(x)$ seja contínuo C^{n-1} (para grau n), o que é essencial para métodos que dependem de gradientes, como a integração das equações de traçamento de raios (Equação 2.4) ou o cálculo do termo físico em PINNs. A eficiência computacional $O(N)$ dos filtros recursivos torna esta abordagem prática para grandes conjuntos de dados em 1D, 2D ou 3D (através de separabilidade) (Unser et al. 1993a).

Capítulo 3

Trabalhos Relacionados

Pesquisadores do mundo todo e de diferentes áreas se dedicam a investigar o tema do traçamento de raios sísmicos e suas aplicações na geofísica computacional. Este capítulo apresenta produções pertencentes ao estado da arte do tema desta pesquisa.

3.1 Pesquisa Bibliográfica

O traçamento de raios sísmicos constitui uma área de pesquisa consolidada na geofísica, com uma extensa literatura que remonta às primeiras aplicações em sismologia e exploração de hidrocarbonetos. Esta seção apresenta os fundamentos teóricos e as definições das ferramentas matemáticas utilizadas.

3.1.1 Fundamentos do Traçamento de Raios Sísmicos

Os fundamentos teóricos do traçamento de raios sísmicos foram estabelecidos por trabalhos fundamentais como o de Virieux et al. (1988), que expandiu a base teórica ao desenvolver métodos específicos para traçamento de raios em meios lateralmente heterogêneos. Posteriormente, Červený (2001) desenvolveu uma teoria abrangente para a propagação de ondas sísmicas em meios heterogêneos e anisotrópicos. Seu livro *Seismic Ray Theory* tornou-se uma referência fundamental, apresentando formulações matemáticas rigorosas baseadas em princípios variacionais e na teoria de Hamiltonianos.

A formulação matemática do problema como um sistema de equações diferenciais ordinárias foi explorada detalhadamente por Červený (1989), que introduziu o conceito de fatoração para meios anisotrópicos. Posteriormente, Margrave (2003) apresentou algoritmos numéricos implementáveis para a solução dessas equações, estabelecendo o arcabouço para os métodos computacionais.

3.1.2 Métodos Numéricos Tradicionais

Os métodos numéricos clássicos evoluíram significativamente nas últimas décadas. Julian (1970) introduziu um dos primeiros algoritmos eficientes para traçamento de raios em meios heterogêneos, baseado em integração numérica direta. Pereyra (1992) desenvolveu o método de “*bending*”, que otimiza iterativamente um caminho inicial para satisfazer

o princípio de Fermat.

Vesnaver (1996) propôs uma abordagem baseada no princípio de Fermat para grades irregulares. O método de Runge-Kutta, amplamente utilizado para integração numérica, foi adaptado para aplicações geofísicas por Margrave & Lamoureaux (2019). Uma contribuição significativa foi feita por Koketsu & Sekine (1998) com o método de “*pseudo-bending*” para uma Terra esférica, enquanto Rawlinson & Sambridge (2004) introduziu o método de *fast marching* para evolução de frentes de onda.

3.1.3 Fundamentos de Redes Neurais e PINNs

Para além dos métodos numéricos clássicos, novas ferramentas computacionais surgiram. As Redes Neurais com Informação Física (PINNs, do inglês *Physics Informed Neural Networks*) emergiram como um *framework* para resolver equações diferenciais. O trabalho pioneiro de Raissi et al. (2019) introduziu o conceito de PINNs para resolver problemas diretos e inversos envolvendo equações diferenciais parciais não-lineares.

Karniadakis et al. (2021) realizaram uma revisão abrangente do aprendizado de máquina com informação física, enquanto Cuomo et al. (2022) analisaram o estado atual e perspectivas futuras das PINNs. Em termos de otimização dessas redes, Wang et al. (2022) demonstraram que o respeito à causalidade é crucial para o treinamento eficaz, e Müller & Zeinhofer (2023) introduziram o *Energy Natural Gradient Descent* (ENGd) para alcançar alta precisão.

3.1.4 Fundamentos das Redes de Kolmogorov-Arnold (KANs)

As Redes de Kolmogorov-Arnold (KANs, do inglês *Kolmogorov-Arnold Networks*) representam uma inovação recente na arquitetura de redes neurais. O trabalho pioneiro de Liu et al. (2025) introduziu as KANs inspiradas pelo teorema da representação de Kolmogorov-Arnold, demonstrando capacidade superior de representação para funções multivariadas. Dhiman (2024) aprofundaram a compreensão teórica das KANs, analisando suas propriedades matemáticas e capacidades de aproximação, estabelecendo a base para seu uso em problemas complexos.

3.2 Estado da Arte

Esta seção analisa como as técnicas descritas anteriormente são aplicadas especificamente no problema de traçamento de raios e modelagem sísmica, delineando o estágio atual do conhecimento e as lacunas persistentes.

3.2.1 Evolução dos Métodos Numéricos em Geofísica

Os métodos numéricos continuam evoluindo para lidar com a complexidade geológica. Virieux & Operto (2009) desenvolveram abordagens para inversão da forma de onda completa, enquanto Červený (1989) introduziu métodos rápidos para meios anisotrópicos baseados em perturbação. A computação de alto desempenho também foi integrada,

com Jaroš et al. (2021) implementando algoritmos em GPUs e Badouel et al. (1990) desenvolvendo *frameworks* paralelos. Contudo, Farra & Madariaga (1988) demonstraram que instabilidades intrínsecas persistem em cenários geologicamente complexos.

3.2.2 Aprendizado de Máquina Aplicado ao Traçamento de Raios

A aplicação de técnicas de aprendizado de máquina transformou o traçamento de raios. Bergen et al. (2019) e Zhu et al. (2019) estabeleceram precedentes na detecção de fases sísmicas. No contexto específico de tempos de trânsito, Smith et al. (2021) desenvolveram a EikoNet para resolver a equação eikonal, demonstrando generalização, mas limitações em fortes contrastes.

O uso de PINNs em geofísica avançou com Moseley et al. (2020a) resolvendo a equação da onda e bin Waheed et al. (2021) aplicando solucionadores neurais para a equação eikonal, abordando patologias causais. Moseley et al. (2020b) aplicaram PINNs à equação da onda sísmica com resultados promissores em modelos sintéticos.

Uma contribuição central para esta discussão é o trabalho de Duarte et al. (2023), que aplicaram PINNs especificamente ao sistema de EDOs do traçamento de raios. Embora tenham demonstrado a viabilidade da abordagem, identificaram limitações significativas na generalização para diferentes condições iniciais, um desafio que motiva novas pesquisas.

3.2.3 Aplicações Recentes de KANs em Física

Embora as KANs sejam recentes, suas aplicações em domínios científicos já começaram a ser exploradas, superando limitações de redes convencionais. Yamak et al. (2025) utilizaram KANs para previsão de séries temporais, enquanto Hadj Kilani (2024) destacaram sua relevância para problemas que exigem alta precisão.

Na fronteira do conhecimento, Shuai & Li (2025) e Patra et al. (2024) realizaram trabalhos pioneiros na integração de KANs com princípios de informação física (Physics-Informed KANs), aplicando-as a sistemas de energia e resolução de equações diferenciais parciais, demonstrando capacidades superiores de generalização e convergência em relação às PINNs tradicionais.

3.3 Contribuições do Trabalho Proposto

A principal contribuição deste trabalho é a introdução pioneira das Redes de Kolmogorov-Arnold com Informação Física (PIKANs) para o problema específico de traçamento de raios sísmicos para múltiplas condições iniciais não vistas durante o treinamento. Enquanto trabalhos anteriores exploraram a aplicação de PINNs convencionais para este problema (Duarte et al. 2023, bin Waheed et al. 2021), a utilização de KANs como arquitetura base representa uma inovação significativa.

As KANs oferecem capacidades superiores de representação para funções multivariadas de maior complexidade, graças à sua fundamentação no teorema da representação de Kolmogorov-Arnold (Liu et al. 2025, Dhiman 2024). Ao integrar esta arquitetura com

princípios de informação física, este trabalho estabelece um novo paradigma para modelagem de propagação de ondas sísmicas.

Uma limitação crítica identificada em trabalhos anteriores é a dificuldade de generalização para diferentes condições iniciais (Duarte et al. 2023, Smith et al. 2021). Tipicamente, as PINNs são treinadas para uma única trajetória de raio com condições iniciais fixas, necessitando retreinamento para cada nova configuração.

A abordagem proposta supera esta limitação ao incorporar condições iniciais (posição da fonte e ângulo de disparo) diretamente como entradas do modelo PIKAN. Esta estratégia, combinada com a capacidade superior de representação das KANs, permite que um único modelo aprenda a mapear famílias contínuas de trajetórias de raios, generalizando efetivamente para condições iniciais não vistas durante o treinamento.

Esta contribuição é particularmente relevante para aplicações práticas em imageamento sísmico e inversão tomográfica, onde múltiplas trajetórias de raios precisam ser calculadas eficientemente. A capacidade de generalização da abordagem proposta representa um avanço significativo em relação ao estado da arte, potencialmente transformando fluxos de trabalho em geofísica computacional.

Outra contribuição deste trabalho é a introdução de uma estratégia de regularização consciente de dados, que pondera adaptativamente a perda física pela densidade local de dados, inspirado pelo trabalho de Eivazi et al. (2022), estendendo este conceito para o contexto específico do traçamento de raios sísmicos, onde a distribuição espacial dos dados de treinamento pode ser altamente não uniforme. A estratégia proposta promove a consistência física em regiões sub-amostradas pelas trajetórias dos raios e o refinamento guiado por dados em regiões bem amostradas.

Capítulo 4

Metodologia

Este capítulo está organizado em três seções, visando detalhar a metodologia proposta: Obtenção do Conjunto de Dados Sintéticos, Treinamento do Modelo e Avaliação da Abordagem. A Figura 4.1 apresenta um esquema geral que sintetiza e orienta a explicação das etapas adotadas na abordagem proposta. O diagrama utiliza a seguinte codificação de cores: processos são representados em preto, entradas dos processos e hiperparâmetros em verde, e saídas em laranja.

Uma característica importante deste trabalho é que todos os experimentos e a implementação dos algoritmos foram realizados na linguagem Python. Para garantir a reprodutibilidade dos resultados, o código-fonte completo está disponível em um repositório público na plataforma GitHub¹, proporcionando acesso aberto às implementações.

4.1 Parte 1: Obtenção do Conjunto de Dados Sintéticos

O objetivo principal desta etapa é gerar um conjunto de dados diversificado e representativo das trajetórias dos raios sísmicos em um meio com um gradiente de velocidade definido por uma nuvem de pontos conhecida, seja por meio de um modelo de velocidade parametrizado ou obtido experimentalmente. Um exemplo de modelo de velocidade esperado é ilustrado na Figura 4.1.

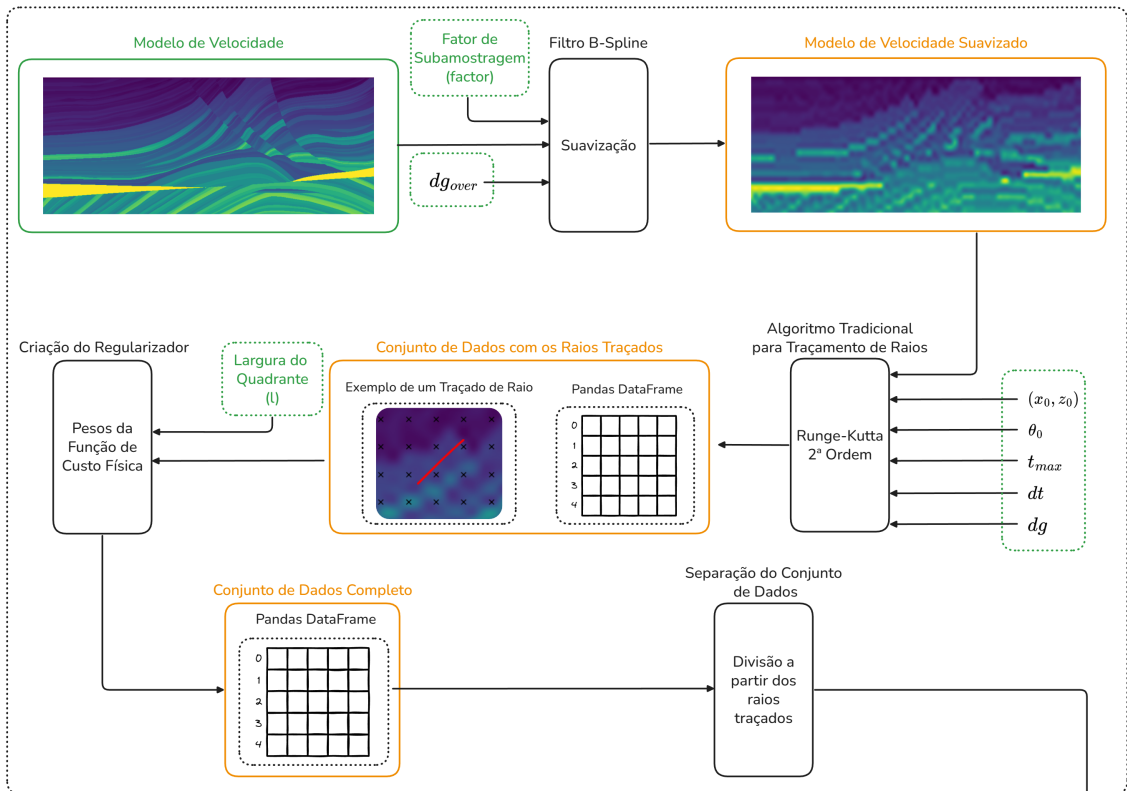
A saída dessa etapa consiste em três conjuntos de dados, utilizados para treinamento, validação e teste do modelo, que contêm, respectivamente, 60%, 20% e 20% do total de raios sísmicos simulados. Nos conjuntos de validação e teste, as variáveis de entrada, ou *features* (características), são: a posição inicial (x_0 e z_0); o ângulo do vetor de vagarosidade inicial (θ_0); e o instante de tempo (t), que define o tempo de propagação do raio desde sua posição inicial em $t = 0$. Já as variáveis de saída, ou *target* (alvo), são: a posição do traçamento do raio (x e z); e as componentes x e z do vetor de vagarosidade (p_x e p_z).

No caso do conjunto de dados de treinamento, as variáveis de entrada e saída são basicamente as mesmas, com a inclusão de cinco variáveis adicionais na entrada, que são utilizadas na função de custo física: as derivadas parciais da posição em relação ao tempo ($\frac{dx}{dt}$ e $\frac{dz}{dt}$); as derivadas parciais do vetor de vagarosidade em relação ao tempo

¹Repositório disponível em: <https://github.com/Lucastmarques/PIKAN-RT>.

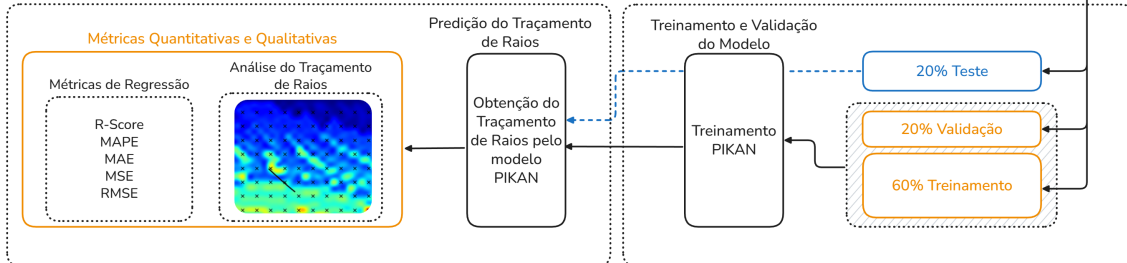
Figura 4.1: Esquema geral da nova abordagem proposta para o traçamento de raios sísmicos. Processos são representados em preto, entradas dos processos e hiperparâmetros em verde, e saídas em laranja.

Parte 1: Obtenção do Conjunto de Dados Sintéticos



Parte 3: Avaliação do Modelo

Parte 2: Treinamento e Validação



Fonte: Elaborado pelo autor (2025).

$\left(\frac{dp_x}{dt} \text{ e } \frac{dp_z}{dt}\right)$; e o peso adaptativo da função de custo física (w_f). A obtenção dessas derivadas e dos pesos w_f serão explicados com mais detalhes nas Subseções 4.1.2 e 4.1.3, respectivamente.

4.1.1 Aplicação do Filtro B-Splines

Como a maioria dos modelos de velocidade possuem descontinuidades, faz-se necessário a suavização do modelo, com o objetivo de garantir a continuidade e diferenciabilidade do campo. Na abordagem proposta, é utilizado a transformada B-splines direta como filtro de suavização, cuja explicação foi apresentada na Seção 2.4 deste trabalho.

A etapa de suavização inicia-se com a subamostragem dos pontos da grade do campo de velocidade por um fator s escolhido, tal que

$$v_{i,j}^{(\text{amostrado})} = v_{i \cdot s, j \cdot s}^{(\text{orig})}, \quad (4.1)$$

onde (j, i) são as coordenadas dos pontos da grade subamostrada do campo de velocidade, e $(i \cdot s, j \cdot s)$ são as coordenadas correspondentes na grade original.

O campo de velocidade subamostrado resultante é definido em uma grade menos precisa, que é representada como uma matriz $\mathbf{I} \in \mathbb{R}^{N \times M}$. Para suavizar este campo discreto, aplica-se a transformada B-splines direta, que opera em duas dimensões aplicando um filtro unidimensional ao longo de cada eixo independentemente. Esta estrutura separável reduz a complexidade computacional e permite uma implementação eficiente (Briand & Monasse 2018, Shulga et al. 2019).

Seja $f : \mathbb{R}^L \rightarrow \mathbb{R}^L$ o operador de filtragem B-splines unidimensional recursivo que calcula os coeficientes splines para uma dada sequência de entrada, onde L representa a dimensão da entrada e da saída do operador de filtragem. Primeiramente, o operador de filtragem f é aplicado a cada linha da imagem $\mathbf{I}$, produzindo uma matriz intermediária $\mathbf{C}^{(0)} \in \mathbb{R}^{N \times M}$. Em seguida, o mesmo operador é aplicado a cada coluna de $\mathbf{C}^{(0)}$ para obter a matriz final de coeficientes B-splines $\mathbf{C} \in \mathbb{R}^{N \times M}$. Este procedimento pode ser formalmente escrito como:

$$\begin{aligned} \mathbf{C}_{i,:}^{(0)} &= f(\mathbf{I}_{i,:}), & \text{para } i = 1, 2, \dots, N, \\ \mathbf{C}_{:,j} &= f(\mathbf{C}_{:,j}^{(0)}), & \text{para } j = 1, 2, \dots, M. \end{aligned} \quad (4.2)$$

Utilizando os coeficientes $c[i, j] \in \mathbf{C}$, é possível reconstruir o campo de velocidade $v(x, z)$, com descontinuidades reduzidas, a partir de uma soma ponderada dos coeficientes próximos. A Equação 4.3 descreve o processo de reconstrução:

$$v(x, z) = \sum_{i=0}^3 \sum_{j=0}^3 c[i+k, j+l] \beta\left(\frac{z}{dg} - k - (i-1)\right) \beta\left(\frac{x}{dg} - l - (j-1)\right), \quad (4.3)$$

onde $k = \lfloor z/dg \rfloor$ e $l = \lfloor x/dg \rfloor$ são as partes inteiras das coordenadas normalizadas, e dg representa o espaçamento entre os coeficientes B-splines vizinhos. Esse espaçamento é

dados pelo produto do fator de subamostragem s e o espaçamento entre as amostras do campo de velocidade discreto dg_{over} , ou seja, $dg = s \cdot dg_{\text{over}}$. O valor de dg_{over} pode ser obtido a partir da frequência de amostragem nos casos em que o modelo de velocidade foi obtido experimentalmente. Por fim, $\beta^3(\cdot)$ é a função de base B-splines cúbica, definida como:

$$\beta^3(x) = \begin{cases} \frac{2}{3} - x^2 + \frac{1}{2}|x|^3, & \text{se } 0 \leq |x| < 1, \\ \frac{(2-|x|)^3}{6}, & \text{se } 1 \leq |x| < 2, \\ 0, & \text{caso contrário.} \end{cases} \quad (4.4)$$

Um exemplo da saída esperada desta etapa é apresentado na Figura 4.1.

4.1.2 Traçamento de Raios Sísmicos

Como explicado na Seção 2.2, as PINNs são tradicionalmente treinadas a partir de um balanceamento entre informações obtidas experimentalmente, isto é, um conjunto de dados observados, e informações obtidas pela teoria, isto é, equações físicas que descrevem e/ou restringem o problema. Desta forma, é necessário obter amostras de traçamentos de raios com diferentes posições iniciais em um mesmo campo de velocidade, para que seja possível obter as informações experimentais para este problema. A metodologia mais utilizada nesses casos é a utilização de dados sintéticos. Esta seção detalha a implementação do método de Runge-Kutta de segunda ordem para a integração numérica das equações canônicas de Hamilton que governam a propagação dos raios em um meio heterogêneo.

Formulação Matemática

Análogo ao que foi apresentado na Seção 2.1, é possível descrever a evolução de um raio sísmico em um meio heterogêneo, para um sistema de coordenadas bidimensionais, pelo seguinte sistema de equações diferenciais ordinárias (EDOs):

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= v^2(\mathbf{x})\mathbf{p}, \\ \frac{d\mathbf{p}}{dt} &= -\frac{1}{v(\mathbf{x})}\nabla v(\mathbf{x}), \end{aligned}$$

onde $\mathbf{x} = (x, z)$ representa a posição espacial do raio, $\mathbf{p} = (p_x, p_z)$ é o vetor de vagarosidade, $v(\mathbf{x})$ é o campo de velocidade, e t é o tempo de trânsito ao longo do raio.

Este sistema pode ser reescrito na forma vetorial como:

$$\frac{d\mathbf{r}}{dt} = \mathbf{f}(\mathbf{r}, t), \quad (4.5)$$

onde $\mathbf{r} = (x, z, p_x, p_z)$ é o vetor de estado do raio e $\mathbf{f}(\mathbf{r}, t)$ é a função que define o lado direito do sistema de EDOs:

$$\mathbf{f}(\mathbf{r}, t) = \left(v^2 p_x, v^2 p_z, -\frac{\partial_x v}{v}, -\frac{\partial_z v}{v} \right). \quad (4.6)$$

Implementação do Método de Runge-Kutta de Segunda Ordem

Para resolver numericamente o sistema de equações diferenciais ordinárias (EDOs) que descrevem a trajetória de um raio sísmico, foi utilizado o método de Runge-Kutta de segunda ordem, também conhecido como método do ponto médio. Este método proporciona um bom equilíbrio entre precisão e eficiência computacional, sendo especialmente adequado para o traçamento de raios sísmicos em modelos geofísicos (Zelt & Ellis 1988, Rawlinson et al. 2008, Zhang et al. 2017).

O algoritmo proposto recebe como entrada o passo de tempo para a integração numérica (dt), o tempo máximo de propagação do raio ($t_{\max}$), o vetor de estado inicial do raio $\mathbf{r}_0 = (x_0, z_0, p_{x0}, p_{z0})$, os limites do domínio computacional nas direções x e z , os coeficientes B-splines do modelo de velocidade ($c[i, j]$) e o espaçamento da grade para esses coeficientes (dg).

A implementação do algoritmo começa com a definição da grade temporal e a alocação de memória para armazenar a trajetória do raio e suas derivadas. Em seguida, os coeficientes do modelo de velocidade são espelhados para garantir que as condições de contorno sejam tratadas adequadamente durante a interpolação B-splines. As derivadas do vetor de estado no instante inicial são calculadas avaliando a função $\mathbf{f}(\mathbf{r}, t)$ no ponto inicial. Tanto o campo de velocidade v quanto as suas derivadas espaciais $\partial_x v$ e $\partial_z v$ são calculadas por meio de interpolação B-splines.

O núcleo do algoritmo consiste na integração numérica, que itera sobre os passos de tempo, aplicando o método de Runge-Kutta de segunda ordem para avançar a solução da trajetória do raio. O Algoritmo 1 ilustra o processo de integração.

O espelhamento simétrico (linha 3 do algoritmo 1) tem o objetivo de estender a matriz de coeficientes replicando valores internos de forma simétrica nas bordas, para garantir a continuidade e valores adequados das derivadas nas regiões da borda do campo de velocidade, conforme explicado na Seção 2.3.

Algoritmo 1: Integração de Trajetória de Raio usando o Método Runge-Kutta de Segunda Ordem

Entrada: Estado inicial do raio $r_0 = (x_0, z_0, p_{x_0}, p_{z_0})$, tempo máximo $t_{\max}$, tamanho do passo de integração dt , coeficientes de velocidade $c[i, j]$, resolução espacial dg , limites do domínio $(x_{\min}, x_{\max})$, $(z_{\min}, z_{\max})$

Saída: Vetor de tempo t , trajetória do raio $r(t)$ e suas derivadas $\dot{r}(t)$

- 1 Calcular número de passos de integração: $N = \lceil t_{\max}/dt \rceil + 1$;
- 2 Inicializar o array de trajetória de raio $r(t) \in \mathbb{R}^{N \times 4}$ e definir $r(0) = r_0$;
- 3 Aplicar espelhamento simétrico nos coeficientes de velocidade:
 $c^{\text{espelhado}} = \text{mirrorW2d}(c)$;
- 4 Calcular a velocidade inicial v_0 e seus gradientes $(\partial_x v, \partial_z v)$ em (x_0, z_0) usando interpolação B-splines;
- 5 Inicializar o array de derivadas $\dot{r}(t) \in \mathbb{R}^{N \times 4}$ e definir:

$$\dot{r}(0) = \left(v_0^2 p_{x_0}, v_0^2 p_{z_0}, -\frac{\partial_x v}{v_0}, -\frac{\partial_z v}{v_0} \right)$$

6 **para** $i = 1$ **até** $N - 1$ **faça**

- 7 Limitar a posição atual (x_i, z_i) para ficar dentro dos limites;
- 8 Avaliar v , $\partial_x v$, e $\partial_z v$ em (x_i, z_i) ;
- 9 Calcular o primeiro passo RK k_1 :

$$k_1 = dt \cdot \left(v^2 p_x, v^2 p_z, -\frac{\partial_x v}{v}, -\frac{\partial_z v}{v} \right)$$

- 10 Estimar o estado no ponto médio $r_{\text{mid}} = r_{i-1} + 0.5 \cdot k_1$;
- 11 Limitar $(x_{\text{mid}}, z_{\text{mid}})$ aos limites;
- 12 Avaliar v , $\partial_x v$, e $\partial_z v$ no ponto médio;
- 13 Calcular o segundo passo RK k_2 :

$$k_2 = dt \cdot \left(v^2 p_x^{\text{mid}}, v^2 p_z^{\text{mid}}, -\frac{\partial_x v}{v}, -\frac{\partial_z v}{v} \right)$$

- 14 Atualizar o estado do raio: $r_i = r_{i-1} + k_2$;
- 15 Avaliar $\dot{r}_i$ como:

$$\dot{r}_i = \left(v^2 p_x^i, v^2 p_z^i, -\frac{\partial_x v}{v}, -\frac{\partial_z v}{v} \right)$$

16 **fim**

- 17 **retorna** $t, r(t), \dot{r}(t)$
-

Em cada iteração, o algoritmo primeiro calcula o incremento k_1 avaliando a função $\mathbf{f}$ no ponto atual. As coordenadas são limitadas aos limites do domínio computacional para evitar extrapolação durante a interpolação do campo de velocidade. Em seguida, estima-se um estado intermediário $\mathbf{r}_{i-1} + 0.5 \cdot k_1$ e calcula-se o incremento k_2 avaliando a função

$\mathbf{f}$ neste ponto médio. O estado do raio é então atualizado utilizando o incremento k_2 , e as derivadas do vetor de estado no ponto atualizado são calculadas e armazenadas para uso posterior.

O método de Runge-Kutta de segunda ordem utilizado nesta implementação pode ser resumido pelas seguintes equações:

$$k_1 = dt \cdot \mathbf{f}(\mathbf{r}_{i-1}, t_{i-1}), \quad (4.7)$$

$$k_2 = dt \cdot \mathbf{f}(\mathbf{r}_{i-1} + 0.5 \cdot k_1, t_{i-1} + 0.5 \cdot dt), \quad (4.8)$$

$$\mathbf{r}_i = \mathbf{r}_{i-1} + k_2. \quad (4.9)$$

Este método oferece uma precisão de segunda ordem ($O(dt^2)$), significativamente melhor que o método de Euler ($O(dt)$), com apenas um estágio intermediário adicional. A função retorna o vetor de tempos de trânsito, a matriz de estados do raio contendo os valores de (x, z, p_x, p_z) para cada instante de tempo, e a matriz de derivadas correspondentes, que são fundamentais para o treinamento do modelo proposto neste trabalho.

4.1.3 Criação do Regularizador

No contexto de redes neurais informadas pela física (PINNs), os pesos da função custo desempenham um papel crucial ao equilibrar a contribuição entre os dados observados e as restrições físicas do modelo. Estudos recentes demonstram que a escolha inadequada desses pesos pode levar a convergência lenta, instabilidade no treinamento ou até mesmo falha completa na aprendizagem (Wang et al. 2021, Xiang et al. 2022). Para garantir um aprendizado adequado ao longo de todo o domínio de interesse, a atribuição dos pesos é realizada de maneira adaptativa, levando em consideração a densidade dos dados em diferentes regiões do espaço de características.

A estratégia adotada baseia-se na frequência espacial dos pontos de dados dentro de intervalos pré-definidos, denominados de quadrantes. Regiões com maior densidade de dados recebem pesos adaptativos menores para a função de custo física (w_f), enquanto regiões com dados mais esparsos são atribuídas com pesos maiores, permitindo que o modelo PINN aprenda de maneira equilibrada e evite o sobreajuste em áreas com muitos dados. Esta abordagem alinha-se com os princípios de balanceamento adaptativo de perdas propostos por (Gao et al. 2025), que demonstraram melhorias significativas na precisão e estabilidade do treinamento quando os pesos são ajustados dinamicamente durante o processo de otimização. Matematicamente, o processo de atribuição dos pesos é descrito a seguir.

Inicialmente, realiza-se a contagem do número de amostras em cada um dos m quadrantes. O número total de quadrantes m é determinado pela largura l — hiperparâmetro da abordagem proposta — e pelas dimensões do modelo de velocidade utilizado. A frequência de dados para cada quadrante, denotada por $\rho(q)$, é definida pelo número de amostras que ele contém.

$$\rho(q) = \sum_{i=1}^n \mathbb{1}[x_i \in [x_{\min}^q, x_{\max}^q] \wedge z_i \in [z_{\min}^q, z_{\max}^q]], \quad (4.10)$$

onde $\mathbb{1}$ é a função indicadora, que assume valor 1 quando o ponto (x_i, z_i) está dentro dos limites $[x_{\min}, x_{\max}]$ e $[z_{\min}, z_{\max}]$, e 0 caso contrário. Os intervalos $[x_{\min}^q, x_{\max}^q]$ e $[z_{\min}^q, z_{\max}^q]$ definem os limites do domínio do quadrante q que está sendo avaliado, respectivamente nos eixos x e z . Enquanto x_i e z_i são os pontos que compõe cada um dos raios traçados no conjunto de dados de treinamento com n amostras.

Após calcular as frequências, estas são normalizadas no intervalo $[0, 1]$ utilizando o escalonamento Min-Max, o que pode ser expresso pela equação:

$$\rho_{\text{norm}}(q) = \frac{\rho(q) - \min(\boldsymbol{\rho})}{\max(\boldsymbol{\rho}) - \min(\boldsymbol{\rho})}, \quad (4.11)$$

onde $\rho_{\text{norm}}(q)$ representa a frequência normalizada do q -ésimo quadrante, e $\min(\boldsymbol{\rho})$ e $\max(\boldsymbol{\rho})$ são os valores mínimo e máximo das frequências calculadas, respectivamente.

Com as frequências normalizadas, os pesos da função de custo física para cada quadrante são calculados como o inverso das frequências normalizadas:

$$w_f(q) = \frac{1}{\rho_{\text{norm}}(q)}. \quad (4.12)$$

Além disso, é imposta uma restrição superior ao valor dos pesos da função de custo física, de modo que $w_f(q)$ seja limitado a um valor máximo de 10, para evitar que regiões com frequências muito baixas tenham um peso excessivo que poderia causar instabilidade no treinamento do modelo.

Atribuindo os pesos da função de custo física aos pontos de dados, para cada ponto (x_i, z_i) , o quadrante correspondente é identificado e o peso da função de custo física correspondente a esse quadrante é atribuído ao ponto. O processo pode ser descrito pela seguinte equação:

$$w_f(x_i, z_i) = w_f(i), \quad \text{se } (x_i, z_i) \in [x_{\min}^q, x_{\max}^q] \times [z_{\min}^q, z_{\max}^q], \quad (4.13)$$

em que $[x_{\min}^q, x_{\max}^q] \times [z_{\min}^q, z_{\max}^q]$ são os valores necessários para a reconstrução de cada quadrante.

Este procedimento garante que os pesos sejam adequadamente distribuídos de acordo com a densidade de dados em cada região do espaço de características.

Algoritmo 2: Cálculo e Atribuição dos Pesos da Função de Custo Física aos Dados

Entrada: Dados (*data*), restrições (*restrictions*), tamanho do passo (*step*)

Saída: Dados com pesos da função de custo física atribuídos

```

1 Definir limites de cada quadrante com base nas restrições;
2 Calcular frequências de dados para cada quadrante usando as restrições e o passo;
3 Normalizar as frequências usando escalonamento Min-Max;
  /* Calcular e Clipar os pesos da função de custo física para
    cada quadrante */
4 para cada quadrante q em restrictions faça
5   |  $w_f(q) = \frac{1}{\rho_{\text{norm}}(q)}$ ;
6   |  $w_f^*(q) = \min(w_f(q), 10)$ 
7 fim
  /* Salvar os pesos para cada registro de data na coluna
    "pi_weight" */
8 para cada  $(x_i, z_i)$  em data faça
9   | Identificar quadrante correspondente q para  $(x_i, z_i)$ ;
10  |  $data[(x_i, z_i), \text{"pi\_weight"}] = w_f^*(q)$ ;
11 fim
  /* Retorna o conjunto de dados com a coluna de pesos da função
    de custo física atribuída */
12 retorna data
```

O Algoritmo 2 mostra como o processo de cálculo e atribuição dos pesos da função de custo para cada amostra do conjunto de dados de treinamento é feito, através da implementação das Equações 4.10 a 4.13. O valor 10 utilizado no *clipping* dos pesos, na linha 6, foi obtido de maneira empírica e é fundamental para evitar que quadrantes com pouco ou nenhum dado receba valores muito grandes, que tendem a infinito.

Por fim, é importante ressaltar que $w_f^*(q)$ é o peso da função de custo física para cada quadrante. Só após a identificação do quadrante e a atribuição dos pesos correspondentes a cada amostra, nas linhas 8 a 11 do Algoritmo 2, que os pesos da função física $w_f(x_i, z_i)$ são obtidos. Estes são utilizados durante o treinamento do modelo, de forma análoga à Equação 2.8, para obter a função de custo total.

4.1.4 Divisão do Conjunto de Dados

Após a simulação de todas as trajetórias, os dados brutos foram organizados em um formato estruturado, especificamente um *DataFrame* da biblioteca Pandas. Cada linha do *DataFrame* corresponde a um ponto específico (x, z) em uma determinada trajetória de raio, definida pelas suas condições iniciais (x_0, z_0, θ_0) , em um instante de tempo t . As colunas do conjunto de dados foram definidas para incluir:

- **Entradas (Features):** As condições iniciais (x_0, z_0, θ_0) que definem o raio e o tempo de trânsito t até o ponto atual.

- **Saídas (Targets):** As variáveis a serem previstas pelo modelo, sendo elas a posição (x, z) e as componentes do vetor de vagarosidade (p_x, p_z) no instante t .
- **Informações Adicionais:** As derivadas temporais $(\frac{dx}{dt}, \frac{dz}{dt}, \frac{dp_x}{dt}, \frac{dp_z}{dt})$ também foram armazenadas, assim como os pesos da função de custo física (w_f) .

Esta estrutura permite que o modelo da abordagem proposta seja treinado para mapear as condições iniciais e o tempo de trânsito para o estado correspondente do raio (posição e vagarosidade).

Para garantir uma avaliação imparcial do desempenho do modelo e evitar o vazamento de informações (do inglês, *data leakage*) entre as fases de treinamento e teste, o conjunto de dados completo foi dividido em três subconjuntos mutuamente exclusivos: Treinamento, Validação (ou Desenvolvimento) e Teste.

No contexto de aprendizado de máquina, a divisão do conjunto de dados em três subconjuntos mutuamente exclusivos segue uma prática metodológica consolidada e essencial para o desenvolvimento de modelos robustos e generalizáveis (Goodfellow et al. 2016). O conjunto de Treinamento é utilizado para ajustar os parâmetros do modelo através do processo de otimização, permitindo que o algoritmo aprenda os padrões subjacentes aos dados. O conjunto de Validação, por sua vez, é utilizado para ajustar hiperparâmetros, selecionar arquiteturas e monitorar o processo de treinamento, evitando o sobreajuste (do inglês, *overfitting*) (Bishop 2006). Já o conjunto de Teste permanece completamente isolado durante todo o processo de desenvolvimento, sendo utilizado apenas uma vez, ao final, para fornecer uma estimativa não-enviesada do desempenho do modelo em dados nunca antes vistos, simulando o comportamento em ambiente de produção (Hastie et al. 2009).

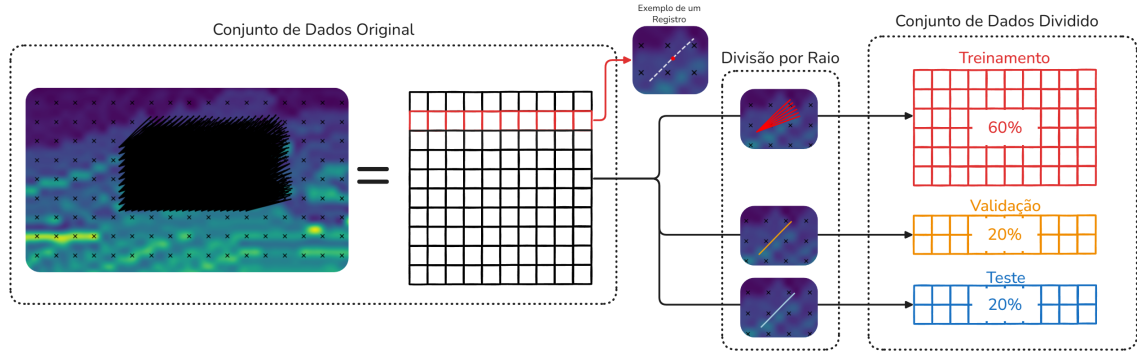
Esta metodologia de divisão tripla também é importante para o problema de traçamento de raios sísmicos, onde a capacidade de generalização para diferentes condições iniciais e modelos de velocidade é fundamental para aplicações práticas em geofísica exploratória. Conforme destacado por (James et al. 2013), a avaliação em um conjunto de teste independente é a única forma confiável de verificar se o modelo aprendeu características genuinamente preditivas ou apenas memorizou peculiaridades dos dados de treinamento.

Assim como mostra a Figura 4.2, a divisão foi realizada estrategicamente com base nas condições iniciais únicas (x_0, z_0, θ_0) que definem cada raio completo. Ou seja, todas as amostras pertencentes a uma mesma trajetória de raio foram mantidas juntas no mesmo subconjunto. Esta abordagem garante que o modelo não seja treinado e testado em segmentos diferentes da mesma trajetória, o que poderia inflar artificialmente as métricas de desempenho. A proporção da divisão (i.e., 60% para treinamento, 20% para validação, 20% para teste) foi aplicada sobre o conjunto de raios únicos.

É crucial ressaltar que esta divisão foi realizada antes de qualquer etapa de pré-processamento adicional (como normalização ou escalonamento dos dados), preservando a independência estatística entre os subconjuntos.

A geração, organização e divisão cuidadosa dos dados sintéticos foram processos fundamentais para assegurar a qualidade e a representatividade do conjunto de dados utilizado no subsequente treinamento e avaliação do modelo PIKAN, sendo, portanto, a última etapa da parte 1 da abordagem proposta.

Figura 4.2: Divisão do Conjunto de Dados original em treinamento, validação e teste.



Fonte: Elaborado pelo autor (2025).

4.2 Parte 2: Treinamento e Validação do Modelo

Assim como todo modelo de Aprendizado de Máquina, o processo de treinamento e validação do modelo híbrido baseado na união de Redes Neurais com Informação Física (PINNs) e Redes de Kolmogorov-Arnold (KANs) constitui uma etapa fundamental para garantir a capacidade preditiva e a conformidade física do sistema proposto. Esta seção detalha a metodologia empregada no treinamento e validação do modelo, abrangendo a formulação matemática da função de perda, as estratégias de otimização, os mecanismos de regularização e o protocolo de validação implementados.

4.2.1 Formulação da Função de Perda

A função de perda total ($\mathcal{L}_{\text{total}}$) foi estruturada para incorporar simultaneamente a aderência aos dados observados e o respeito às leis físicas que governam o fenômeno modelado. Conforme proposto por (Raissi et al. 2019), a abordagem de PINNs fundamenta-se na combinação ponderada de termos de perda supervisionada e restrições físicas. No contexto deste trabalho, a função de perda total é definida como:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_d + \lambda_f \mathcal{L}_f + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}} \quad (4.14)$$

onde $\mathcal{L}_d$ representa a perda associada aos dados observados, $\mathcal{L}_f$ corresponde à perda relacionada às restrições físicas, λ_f é o peso atribuído à função de perda física, e $\mathcal{L}_{\text{reg}}$ engloba os termos de regularização propostos por (Liu et al. 2025).

Perda de Dados

A perda de dados quantifica a discrepância entre as previsões do modelo e os valores observados. Neste trabalho, adotou-se o erro quadrático médio (MSE) como métrica de avaliação, alinhando-se com as recomendações de (Goodfellow et al. 2016) para problemas de regressão:

$$\mathcal{L}_d = \frac{1}{N} \sum_{i=1}^N \|\hat{\mathbf{y}}_i - \mathbf{y}_i\|_2^2, \quad (4.15)$$

onde N é o número de amostras do conjunto de dados de treinamento, $\hat{\mathbf{y}}_i$ representa a saída prevista pelo modelo para a entrada $\mathbf{x}_i$, e $\mathbf{y}_i$ corresponde ao valor observado.

Perda Física

A perda física incorpora o conhecimento das leis que governam o sistema, garantindo que as soluções geradas pelo modelo respeitem os princípios físicos subjacentes. No contexto do traçamento de raios sísmicos, esta perda é formulada a partir das equações canônicas de Hamilton (Červený 2001):

$$\mathcal{L}_f = \frac{1}{N} \sum_{i=1}^N \left(\left\| \frac{d\mathbf{x}_i}{dt} - v^2(\mathbf{x}_i) \mathbf{p}_i \right\|_2^2 + \left\| \frac{d\mathbf{p}_i}{dt} + \frac{\nabla v(\mathbf{x}_i)}{v(\mathbf{x}_i)} \right\|_2^2 \right), \quad (4.16)$$

onde $\mathbf{x}_i$ representa a posição do raio, $\mathbf{p}_i$ é o vetor de vagarosidade, $v(\mathbf{x}_i)$ é a velocidade no ponto $\mathbf{x}_i$, e $\nabla v(\mathbf{x}_i)$ é o gradiente da velocidade.

A implementação da perda física requer o cálculo de derivadas parciais, que é realizado através de diferenciação automática, conforme destacado por (Baydin et al. 2018). Esta abordagem permite a propagação eficiente dos gradientes através da rede neural, facilitando a incorporação das restrições físicas no processo de otimização.

Regularização

Para mitigar o sobreajuste e melhorar a generalização do modelo, foram implementados múltiplos termos de regularização, seguindo as diretrizes estabelecidas para Redes de Kolmogorov-Arnold (Liu et al. 2025). A perda de regularização é definida como:

$$\mathcal{L}_{\text{reg}} = \lambda_{L1} \mathcal{L}_{L1} + \lambda_{\text{ent}} \mathcal{L}_{\text{ent}} + \lambda_{\text{coef}} \mathcal{L}_{\text{coef}} + \lambda_{\text{coefdiff}} \mathcal{L}_{\text{coefdiff}}, \quad (4.17)$$

onde $\mathcal{L}_{L1}$ é a regularização L1 aplicada aos parâmetros da rede, promovendo esparsidade (Tibshirani 2018); $\mathcal{L}_{\text{ent}}$ é a regularização baseada em entropia, que incentiva distribuições não-uniformes dos pesos; $\mathcal{L}_{\text{coef}}$ penaliza a magnitude dos coeficientes das funções splines; $\mathcal{L}_{\text{coefdiff}}$ penaliza as variações bruscas entre coeficientes adjacentes, promovendo suavidade nas funções aprendidas.

Os hiperparâmetros λ_{L1} , λ_{ent} , λ_{coef} e $\lambda_{\text{coefdiff}}$ controlam a influência de cada termo de regularização na função de perda total.

No caso geral de qualquer rede neural, a regularização L1 consiste em acrescentar à função de perda original um termo proporcional à soma dos valores absolutos de todos os parâmetros do modelo:

$$\mathcal{L}_1(\theta) = \|\theta\|_1 = \sum_{k=1}^P |\theta_k|. \quad (4.18)$$

No contexto específico das Redes de Kolmogorov–Arnold (KAN), a regularização L1 penaliza a magnitude média das próprias funções de ativação. Conforme definido (Liu et al. 2025), a norma L1 de uma única função de ativação ϕ denotada por $|\phi|_1$, é sua magnitude média sobre um conjunto de N_p entradas:

$$\mathcal{L}_1 = |\phi|_1 = \frac{1}{N_p} \sum_{i=1}^{N_p} |\phi(x_i)| \quad (4.19)$$

Um valor baixo de $|\phi|_1$ indica que a função contribui pouco para a saída da rede, tornando-a uma excelente opção para ser removida no processo de simplificação da rede, conhecido por poda (do inglês, *pruning*). Aplicando este conceito a uma camada Φ de um modelo baseado na arquitetura KAN, com n_{in} entradas e n_{out} saídas, o termo de regularização L1 se torna a soma das normas L1 de todas essas funções:

$$\mathcal{L}_{\text{KAN-L1}} = \sum_{i=1}^{n_{\text{in}}} \sum_{j=1}^{n_{\text{out}}} |\phi_{i,j}|_1 \quad (4.20)$$

No contexto das KANs, a regularização L1 e a regularização por entropia atuam em sinergia para aumentar a esparsidade da rede. Enquanto a L1 encolhe a magnitude de todas as funções de ativação, a regularização por entropia incentiva uma distribuição de importância pouco uniforme (de baixa entropia), forçando a rede a concentrar sua expressividade em poucas conexões vitais. Desta forma, a entropia de Φ é definida por (Liu et al. 2025):

$$\mathcal{L}_{\text{ent}} = - \sum_{i=1}^{n_{\text{in}}} \sum_{j=1}^{n_{\text{out}}} p_e \log p_e, \quad p_e = \frac{|\phi_{i,j}|_1 + \varepsilon}{|\Phi| + \varepsilon}, \quad (4.21)$$

onde ε é um valor infinitesimal que evita divisões por zero.

4.2.2 Arquitetura e Implementação

A implementação do modelo híbrido PIKAN foi realizada utilizando o *framework* PyTorch, que oferece suporte nativo à diferenciação automática e facilita a implementação de arquiteturas personalizadas (Paszke et al. 2019). A estrutura do sistema foi encapsulada em duas classes principais: `Architecture` e `MappedLoss`.

A classe `Architecture` gerencia o modelo KAN, as funções de perda, os otimizadores, os agendadores — responsáveis por atualizar a taxa de aprendizagem dos otimizadores de acordo com alguma estratégia definida antes do treinamento — e as funções físicas. Esta abordagem modular facilita a experimentação com diferentes configurações e hiperparâmetros (Smith 2018). A classe implementa métodos essenciais como `train_step_fn`, `val_step_fn`, `set_loaders` e `train`, que orquestram o processo de treinamento e validação.

A classe `MappedLoss`, por sua vez, encapsula os diferentes componentes da função de perda, permitindo o monitoramento individualizado de cada termo durante o treinamento. Esta abordagem facilita a análise da contribuição relativa de cada componente e auxilia na identificação de possíveis desequilíbrios (Wang et al. 2021).

4.2.3 Processo de Treinamento

O treinamento do modelo híbrido PIKAN foi conduzido utilizando mini-lotes (*mini-batches*), uma técnica que equilibra eficiência computacional e estabilidade de convergência (Goodfellow et al. 2016). O processo de treinamento pode ser descrito pelo Algoritmo 3.

Algoritmo 3: Treinamento do Modelo Híbrido PIKAN

Entrada: Conjunto de dados de treino ($\mathcal{D}_{\text{treino}}$), conjunto de dados de validação ($\mathcal{D}_{\text{val}}$), número de épocas ($n_{\text{épocas}}$), parâmetros de regularização (λ), parâmetros de atualização da grade (θ_{grade})

Saída: Modelo treinado $\mathcal{M}$ e histórico de perdas

```

1 Inicializar modelo KAN  $\mathcal{M}$ , otimizador  $O$  e agendador  $\mathcal{S}$ ;
2  $B \leftarrow |\mathcal{D}_{\text{treino}}|$ ;
3 para época = 1 até  $n_{\text{épocas}}$  faça
4    $\mathcal{L}_{\text{treino}} \leftarrow 0$ ;
5   para cada mini-lote  $(X_{\text{batch}}, Y_{\text{batch}})$  em  $\mathcal{D}_{\text{treino}}$  faça
6     se critério de atualização da grade for satisfeito então
7       | Atualizar parâmetros da grade  $\theta_{\text{grade}}$ ;
8     fim
9      $\hat{Y}_{\text{batch}} \leftarrow \mathcal{M}(X_{\text{batch}})$ ;
10     $\mathcal{L}_d \leftarrow \text{MSE}(\hat{Y}_{\text{batch}}, Y_{\text{batch}})$ ;
11     $\mathcal{L}_f \leftarrow \text{PhysicsLoss}(\mathcal{M}, X_{\text{batch}})$ ;
12     $\mathcal{L}_{\text{reg}} \leftarrow \text{RegLoss}(\mathcal{M}, \lambda)$ ;
13     $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_d + \lambda_f \mathcal{L}_f + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}}$ ;
14    Calcular gradientes  $\nabla_{\theta} \mathcal{L}_{\text{total}}$ ;
15    Atualizar parâmetros:  $\theta \leftarrow \theta - \alpha \nabla_{\theta} \mathcal{L}_{\text{total}}$ ;
16     $\mathcal{L}_{\text{treino}} \leftarrow \mathcal{L}_{\text{treino}} + \mathcal{L}_{\text{total}}$ ;
17  fim
18   $\mathcal{L}_{\text{treino}} \leftarrow \frac{\mathcal{L}_{\text{treino}}}{B}$ ;
19   $\mathcal{L}_{\text{val}} \leftarrow \text{Validar}(\mathcal{M}, \mathcal{D}_{\text{val}})$ ;
20  Atualizar agendador  $\mathcal{S}$  com base em  $\mathcal{L}_{\text{val}}$ ;
21  Registrar métricas de desempenho;
22 fim
23 retorna Modelo  $\mathcal{M}$  treinado e histórico de perdas

```

Um aspecto distintivo do treinamento de modelos KAN é a atualização adaptativa da grade de splines, que permite refinar a representação em regiões de maior complexidade. Conforme proposto por (Liu et al. 2025), a grade é atualizada periodicamente durante o treinamento, com base na distribuição dos dados de entrada. Este processo é controlado por parâmetros como `update_grid`, `n_grid_update`, `start_grid_update_step` e `stop_grid_update_step`, que determinam quando e com que frequência as atualizações ocorrem. A seguir, apresenta-se uma formulação matemática formal que descreve

como esses parâmetros interagem para controlar o processo de atualização:

Seja $\mathcal{G}$ a grade de splines do modelo KAN, t a época de treinamento atual, e $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ o conjunto de dados de treinamento. O processo de atualização da grade é governado pelos seguintes parâmetros:

- $\delta \in \{0, 1\}$: parâmetro binário `update_grid` que habilita ($\delta = 1$) ou desabilita ($\delta = 0$) o mecanismo de atualização adaptativa;
- $n_g \in \mathbb{N}^+$: parâmetro `n_grid_update` que define o número total de atualizações a serem realizadas durante o treinamento;
- $t_{\text{start}} \in \mathbb{N}$: parâmetro `start_grid_update_step` que define o passo de treinamento a partir do qual as atualizações começam;
- $t_{\text{stop}} \in \mathbb{N}^+$: parâmetro `stop_grid_update_step` que define o passo de treinamento após o qual as atualizações cessam.

A frequência de atualização f_{update} é determinada por

$$f_{\text{update}} = \left\lfloor \frac{t_{\text{stop}} - t_{\text{start}}}{n_g} \right\rfloor \quad (4.22)$$

A condição para atualização da grade no passo t é dada por:

$$\text{Atualizar}(\mathcal{G}, t) = \beta \left[t_{\text{start}} \leq t < t_{\text{stop}} \right] \left[t - t_{\text{start}} \equiv 0 \pmod{f_{\text{update}}} \right], \quad (4.23)$$

Quando $\text{Atualizar}(\mathcal{G}, t) = 1$, a grade $\mathcal{G}$ é refinada com base na distribuição dos dados de entrada $\{x_i\}_{i=1}^N$. O processo de refinamento adapta a densidade dos nós da grade para concentrar mais recursos computacionais em regiões do espaço de entrada com maior densidade de dados ou maior complexidade funcional.

Formalmente, para cada dimensão d do espaço de entrada, a densidade de nós $\rho_d(x)$ na posição x é ajustada proporcionalmente à densidade local dos dados $p_d(x)$:

$$\rho_d(x) \propto p_d(x)^\alpha, \quad (4.24)$$

onde α é um parâmetro que controla o grau de adaptatividade. Este processo resulta em uma grade não uniforme que aloca mais nós em regiões de maior interesse, otimizando assim a capacidade de representação do modelo KAN.

A implementação prática deste mecanismo utiliza o método `update_grid_from_samples` da classe KAN, que recebe as amostras do mini-lote atual e atualiza a grade com base na distribuição dessas amostras. O código verifica a condição de atualização a cada passo de treinamento:

$$\text{if } (t \bmod f_{\text{update}} = 0) \text{ and } (t < t_{\text{stop}}) \text{ and } (t \geq t_{\text{start}}) \text{ and } (\beta = 1) \text{ then update_grid} \quad (4.25)$$

Este mecanismo de atualização adaptativa permite que o modelo KAN ajuste dinamicamente sua capacidade de representação durante o treinamento, concentrando recursos

em regiões do espaço de entrada que requerem maior precisão, resultando em melhor desempenho e eficiência computacional.

A otimização dos parâmetros do modelo foi realizada utilizando o algoritmo Adam (Kingma & Ba 2017), que adapta as taxas de aprendizado para cada parâmetro individualmente, facilitando a convergência em problemas não convexos. A escolha deste otimizador é particularmente adequada para o treinamento de modelos híbridos PIKAN, devido à natureza heterogênea dos gradientes provenientes dos diferentes termos da função de perda.

4.2.4 Ajuste do Tamanho da Grade na Rede de Kolmogorov-Arnold

Durante o treinamento do modelo KAN, alguns experimentos envolvendo a otimização de hiperparâmetros foram realizados, sendo o principal o ajuste do tamanho da grade na arquitetura PIKAN, com o objetivo de identificar o valor de G que minimiza o erro de validação do modelo. Formalmente:

$$G^* = \arg \min_{G \in \mathcal{G}} \mathcal{L}_{\text{val}}(\mathcal{M}_{\theta_G}), \quad (4.26)$$

onde $\mathcal{G} \in \mathbb{N}^+$ é o conjunto de tamanhos de grade considerados, $\mathcal{M}_{\theta_G}$ representa o modelo PIKAN treinado com tamanho de grade G , e $\mathcal{L}_{\text{val}}$ é a função que retorna as métricas avaliadas no conjunto de validação.

Refinamento Progressivo do Tamanho da Grade

A metodologia adotada consiste no refinamento progressivo da grade sobre aprendizado contínuo em KANs (Liu et al. 2025). Este processo pode ser formalizado como uma sequência de problemas de otimização:

$$\theta_{G_{i+1}} = \arg \min_{\theta} \mathcal{L}(\theta) \quad \text{sujeito a} \quad \mathcal{P}_{G_i \rightarrow G_{i+1}}(\theta_{G_i}) = \theta, \quad (4.27)$$

onde G_i e G_{i+1} são tamanhos de grade consecutivos; na sequência $\mathcal{G}$, θ_{G_i} são os parâmetros otimizados para o tamanho de grade G_i , e $\mathcal{P}_{G_i \rightarrow G_{i+1}}$ é um operador de projeção que mapeia os parâmetros de um modelo com tamanho de grade G_i para um modelo equivalente com tamanho de grade G_{i+1} .

O operador de projeção $\mathcal{P}_{G_i \rightarrow G_{i+1}}$ preserva a função representada pelo modelo original através da interpolação dos coeficientes B-splines. Para cada função unidimensional g parametrizada por coeficientes $\{c_j\}_{j=1}^{n_i}$ em uma grade de tamanho G_i , os novos coeficientes $\{c'_j\}_{j=1}^{n_{i+1}}$ para uma grade de tamanho G_{i+1} são calculados de modo a minimizar:

$$\int_a^b \left| \sum_{j=1}^{n_i} c_j B_j^k(x) - \sum_{j=1}^{n_{i+1}} c'_j B_j^k(x) \right|^2 dx \quad (4.28)$$

Este processo de refinamento progressivo permite que o modelo preserve o conhecimento adquirido em grades menores enquanto ganha capacidade de representação adicional com o aumento do tamanho da grade.

Protocolo Experimental

O protocolo experimental para o ajuste do tamanho da grade seguiu uma abordagem sistemática estruturada nas seguintes etapas:

1. **Inicialização:** O modelo PIKAN foi inicializado com o menor tamanho de grade e parâmetros aleatórios θ_{G_1} seguindo a inicialização proposta por Liu et al. (2025):

$$c_{l,i,j} \sim N\left(0, \frac{2}{d_{l-1} \cdot n_l}\right), \quad w_{l,i,j} \sim N\left(0, \frac{2}{d_{l-1}}\right) \quad (4.29)$$

2. **Treinamento inicial:** O modelo foi treinado por um número específico de épocas² utilizando o otimizador Adam com taxa de aprendizado $\eta = 10^{-3}$ e tamanho de lote $B = 128$:

$$\theta_{G_1}^{(t+1)} = \theta_{G_1}^{(t)} - \eta \cdot \text{Adam}\left(\nabla_{\theta} \mathcal{L}(\theta_{G_1}^{(t)})\right) \quad (4.30)$$

3. **Avaliação:** O modelo treinado foi avaliado no conjunto de validação, calculando as métricas de desempenho³ para cada variável alvo.
4. **Refinamento progressivo:** Para cada tamanho de grade subsequente $G_i \in \mathbb{N}^+$, o modelo foi refinado e treinado:

$$\theta_{G_i}^{(0)} = \mathcal{P}_{G_{i-1} \rightarrow G_i}(\theta_{G_{i-1}}) \quad (4.31)$$

$$\theta_{G_i}^{(t+1)} = \theta_{G_i}^{(t)} - \eta \cdot \text{Adam}\left(\nabla_{\theta} \mathcal{L}(\theta_{G_i}^{(t)})\right), \quad t = 0, 1, \dots, E_i - 1 \quad (4.32)$$

onde E_i é o número de épocas de treinamento para cada configuração.

5. **Análise comparativa:** Os resultados de todas as configurações foram analisados para identificar o tamanho de grade ótimo G^* que minimiza o erro de validação:

$$G^* = \arg \min_{G \in \mathcal{G}} \frac{1}{4} \sum_{j=1}^4 \text{RMSE}(\mathbf{y}_j, \hat{\mathbf{y}}_j) \quad (4.33)$$

onde $\mathbf{y}_j$ e $\hat{\mathbf{y}}_j$ representam os valores reais e previstos para a j -ésima variável alvo, respectivamente.

Este protocolo experimental permitiu uma avaliação sistemática do impacto do tamanho da grade na capacidade de representação e no desempenho preditivo do modelo PIKAN, fornecendo mais informações sobre o balanço entre complexidade do modelo e generalização.

²Para mais informações sobre os experimentos, consultar o Capítulo 5.

³Para mais informações sobre as métricas de avaliação, consultar a Parte 4.3.5.

4.2.5 Ajuste do Fator de Subamostragem no Pré-processamento do Campo de Velocidade

O fator de subamostragem s é um hiperparâmetro crítico no pré-processamento do campo de velocidade, determinando o nível de suavização aplicado ao modelo Marmousi através da interpolação B-splines. O objetivo deste ajuste é encontrar o valor ideal de s que equilibre a mitigação de descontinuidades numéricas com a preservação das características geológicas essenciais do modelo. Formalmente, busca-se o fator que otimiza o desempenho do modelo PIKAN no conjunto de validação:

$$s^* = \arg \min_{s \in \mathcal{F}} \mathcal{L}_{\text{val}}(\mathcal{M}_{\theta_s}), \quad (4.34)$$

onde $\mathcal{F} \in \mathbb{N}^+$ é o conjunto de fatores de subamostragem considerados, $\mathcal{M}_{\theta_s}$ representa o modelo PIKAN treinado com o campo de velocidade pré-processado com o fator s , e $\mathcal{L}_{\text{val}}$ é a função que retorna as métricas avaliadas no conjunto de validação.

Protocolo Experimental

O protocolo experimental para o ajuste do fator de subamostragem seguiu uma abordagem sistemática estruturada nas seguintes etapas:

1. **Seleção dos Fatores:** Foi selecionado um conjunto discreto de fatores de subamostragem $\mathcal{F} = \{20, 25, 30, 35, 40\}$ para avaliação.
2. **Pré-processamento do Campo de Velocidade:** Para cada $s \in \mathcal{F}$, o campo de velocidade do modelo Marmousi foi subamostrado e interpolado usando B-splines, gerando um campo de velocidade suavizado.
3. **Treinamento e Avaliação:** O modelo PIKAN foi treinado por 10 épocas para cada campo de velocidade pré-processado, utilizando o tamanho de grid de $G = 4$, otimizador Adam com taxa de aprendizado $\eta = 10^{-3}$ e tamanho de lote $B = 128$.
4. **Avaliação Quantitativa e Qualitativa:** O modelo treinado foi avaliado no conjunto de validação, calculando as métricas de desempenho⁴ para cada variável alvo. Adicionalmente, foi realizada uma análise qualitativa da preservação das estruturas geológicas no campo de velocidade suavizado.
5. **Análise Comparativa:** Os resultados de todas as configurações foram analisados para identificar o fator de subamostragem ótimo s^* que minimiza o erro de validação e maximiza a fidelidade estrutural.

Este protocolo experimental permite uma avaliação sistemática do impacto do fator de subamostragem na qualidade do campo de velocidade e, conseqüentemente, no desempenho preditivo e na estabilidade do modelo PIKAN.

4.2.6 Validação do Modelo

A validação do modelo durante o treinamento é essencial para monitorar o progresso, evitar o sobreajuste e ajustar hiperparâmetros. Neste trabalho, adotou-se uma abordagem

⁴Para mais informações sobre as métricas de avaliação, consultar a Seção 4.3.

de validação contínua, avaliando o desempenho do modelo em um conjunto de dados independente após cada época de treinamento.

O processo de validação segue uma metodologia similar à proposta por (Hastie et al. 2009), onde o modelo é avaliado em termos de sua capacidade de generalização para dados não vistos durante o treinamento. A função `val_step_fn` implementa este processo, calculando a perda de dados no conjunto de validação sem atualizar os parâmetros do modelo.

Um aspecto importante da validação é o uso de agendadores para atualização da taxa de aprendizado, que ajustam dinamicamente o passo de otimização com base no desempenho do modelo no conjunto de validação. Conforme destacado por (Loshchilov & Hutter 2017), esta abordagem pode melhorar significativamente a convergência e o desempenho final do modelo. No presente trabalho, utilizou-se um agendador do tipo `ReduceLROnPlateau`, fornecido pelo *framework* PyTorch, que reduz a taxa de aprendizado quando o desempenho no conjunto de validação estagna.

Para facilitar a análise do processo de treinamento e validação, foram implementadas funções de visualização como `plot_losses`, que geram gráficos das diferentes componentes da perda ao longo das épocas. Esta abordagem visual permite identificar rapidamente problemas como sobreajuste, subajuste ou desequilíbrios entre os diferentes termos da função de perda (Goodfellow et al. 2016).

Por fim, é importante ressaltar que a perda física $\mathcal{L}_f$ e as regularizações $\mathcal{L}_{\text{reg}}$ só foram calculadas para o conjunto de dados de treinamento, uma vez que sua função era exclusivamente para auxiliar a convergência do modelo. Desta forma, os gráficos gerados por `plot_losses` se restringem à função de custo física, função de custo dos dados e a função de custo total para o conjunto de dados de treinamento. Já para o conjunto de dados de validação, tem-se apenas a função de custo dos dados. Ou seja, apenas a função de custo dos dados pode e deve ser avaliada, a partir do conjunto de validação, durante o treinamento do modelo.

4.2.7 Outras Estratégias de Regularização

Além dos termos de regularização incluídos na função de perda, foram implementadas estratégias adicionais para melhorar a robustez e a generalização do modelo. Uma dessas estratégias é o uso de pesos adaptativos para a função de perda física (Wang et al. 2021). Esta abordagem ajusta dinamicamente o peso w_f com base na densidade dos dados em diferentes regiões do espaço de características, conforme explicado na Subseção 4.1.3, permitindo um equilíbrio mais adequado entre a aderência aos dados e o respeito às restrições físicas.

A técnica de *singularity avoiding*, uma estratégia proposta e implementada pelos autores do trabalho original da KAN (Liu et al. 2025), também foi utilizada no modelo. Ela visa mitigar problemas associados a singularidades nas funções splines que compõem a arquitetura, modificando ligeiramente a forma como as funções são avaliadas em regiões próximas a descontinuidades e melhorando a estabilidade numérica do modelo.

Formalmente, considere uma função splines $S(x)$ definida por segmentos em uma grade de nós $\{x_i\}_{i=0}^n$. Em uma implementação padrão, quando a entrada x se aproxima

de um nó x_i , podem ocorrer instabilidades numéricas devido à transição entre diferentes segmentos polinomiais. A técnica de *singularity avoiding* introduz uma função de suavização $\phi_\epsilon(x, x_i)$ que modifica o comportamento da avaliação da splines nas proximidades dos nós:

$$\tilde{S}(x) = \begin{cases} S(x_i + \epsilon \cdot \text{sgn}(x - x_i)), & \text{se } |x - x_i| < \epsilon \text{ para algum } i \\ S(x), & \text{caso contrário} \end{cases} \quad (4.35)$$

onde $\epsilon > 0$ é um parâmetro de tolerância pequeno (tipicamente $\epsilon \approx 10^{-6}$), e $\text{sgn}(\cdot)$ é a função sinal:

$$\text{sgn}(z) = \begin{cases} -1, & \text{se } z < 0 \\ 0, & \text{se } z = 0 \\ 1, & \text{se } z > 0 \end{cases} \quad (4.36)$$

Esta modificação garante que a avaliação da função nunca ocorra exatamente nos nós da grade, mas sim em pontos ligeiramente deslocados, evitando assim potenciais singularidades. Para funções splines multidimensionais $\mathbf{S}(\mathbf{x})$, onde $\mathbf{x} \in \mathbb{R}^d$, a técnica é aplicada componente a componente:

$$\tilde{\mathbf{S}}(\mathbf{x})_j = \tilde{S}_j(x_1, x_2, \dots, x_d), \quad (4.37)$$

para cada componente j da função vetorial.

Durante o treinamento, quando o parâmetro `singularity_avoiding` é definido como `True`, todas as avaliações diretas do modelo KAN utilizam esta versão modificada das funções splines. Isto é particularmente importante durante a retropropagação, onde o cálculo de gradientes pode ser severamente afetado por instabilidades numéricas próximas às descontinuidades das derivadas das funções splines.

A implementação desta técnica resulta em uma melhoria significativa na estabilidade do treinamento, especialmente em modelos KAN profundos ou quando as funções alvo apresentam variações abruptas. Matematicamente, esta abordagem preserva a continuidade da função enquanto evita os pontos críticos onde as derivadas podem apresentar comportamento singular.

4.2.8 Considerações sobre Reprodutibilidade

Para garantir a reprodutibilidade dos resultados, implementou-se um mecanismo de controle de sementes aleatórias através da função `set_seed`, que configura as sementes para todos os geradores de números pseudoaleatórios utilizados no processo de treinamento. Esta prática está alinhada com as recomendações de (Pineau et al. 2021) para pesquisa reproduzível em aprendizado de máquina.

Adicionalmente, foram implementadas funcionalidades para salvar e carregar *checkpoints* do modelo através dos métodos `save_checkpoint` e `load_checkpoint`. Estas funcionalidades permitem interromper e retomar o treinamento, facilitando experimentos de longa duração e possibilitando a comparação controlada entre diferentes configurações.

4.3 Parte 3: Avaliação do Modelo

A avaliação sistemática e rigorosa de modelos baseados em aprendizado de máquina constitui uma etapa fundamental no desenvolvimento de soluções computacionais para problemas físicos complexos (Goodfellow et al. 2016). Esta seção apresenta a metodologia empregada para avaliar o desempenho do modelo PIKAN (*Physics-Informed Kolmogorov-Arnold Network*) proposto neste trabalho, abrangendo tanto aspectos quantitativos quanto qualitativos.

4.3.1 Métricas de Avaliação Quantitativa

A natureza multivariada do problema de traçamento de raios sísmicos, que envolve a predição simultânea de quatro variáveis de saída (posição x , z e componentes do vetor de vagarosidade p_x , p_z), exige uma abordagem de avaliação que considere o desempenho do modelo em cada dimensão separadamente (Hastie et al. 2009). Para tanto, foram implementadas cinco métricas complementares que capturam diferentes aspectos da qualidade preditiva do modelo, sendo elas: O Coeficiente de Determinação; o Erro Absoluto Médio; o Erro Percentual Absoluto Médio; o Erro Quadrático Médio; e a Raiz do Erro Quadrático Médio.

Coeficiente de Determinação (R^2)

O coeficiente de determinação, ou R^2 , quantifica a proporção da variância na variável dependente que é previsível a partir das variáveis independentes (Draper & Smith 1966). Para um problema de regressão multivariada, o R^2 é calculado separadamente para cada variável de saída, conforme a equação:

$$R_j^2 = 1 - \frac{\sum_{i=1}^n (y_{ij} - \hat{y}_{ij})^2}{\sum_{i=1}^n (y_{ij} - \bar{y}_j)^2}, \quad (4.38)$$

onde y_{ij} representa o valor observado da j -ésima variável de saída para a i -ésima amostra, $\hat{y}_{ij}$ é o valor predito correspondente, $\bar{y}_j$ é a média dos valores observados para a j -ésima variável, e n é o número total de amostras. O R^2 varia entre $(-\infty, 1]$, onde 1 indica uma previsão perfeita, 0 indica que o modelo não é melhor que um modelo constante, e valores negativos indicam um desempenho pior que um modelo constante.

Esta métrica foi escolhida por sua interpretabilidade e capacidade de quantificar a proporção da variabilidade explicada pelo modelo, sendo amplamente utilizada na avaliação de modelos físicos (Wang et al. 2021).

Erro Absoluto Médio (MAE)

O MAE mede a magnitude média dos erros em um conjunto de previsões, sem considerar sua direção. É calculado como a média das diferenças absolutas entre os valores previstos e observados:

$$\text{MAE}_j = \frac{1}{n} \sum_{i=1}^n |y_{ij} - \hat{y}_{ij}| \quad (4.39)$$

Esta métrica é robusta a *outliers* e fornece uma medida diretamente interpretável na mesma unidade das variáveis originais (Chai & Draxler 2014). No contexto do traçamento de raios sísmicos, o MAE oferece uma indicação clara do erro médio esperado nas coordenadas espaciais e nas componentes do vetor de vagarosidade.

Erro Percentual Absoluto Médio (MAPE)

O MAPE expressa a precisão como uma porcentagem do erro, oferecendo uma medida relativa e adimensional que facilita a comparação entre diferentes variáveis de saída (Hyndman & Koehler 2006). É calculado como:

$$\text{MAPE}_j = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_{ij} - \hat{y}_{ij}}{y_{ij}} \right| \times 100\% \quad (4.40)$$

Esta métrica é particularmente útil para avaliar o desempenho do modelo em termos relativos, permitindo comparações entre variáveis com diferentes escalas. No entanto, é importante notar que o MAPE pode ser problemático quando os valores observados se aproximam de zero, resultando em divisões instáveis (Makridakis 1993). No contexto do traçamento de raios sísmicos, esta limitação foi considerada durante a análise dos resultados.

Erro Quadrático Médio (MSE)

O MSE é uma das métricas mais utilizadas em problemas de regressão, calculando a média dos quadrados dos erros:

$$\text{MSE}_j = \frac{1}{n} \sum_{i=1}^n (y_{ij} - \hat{y}_{ij})^2 \quad (4.41)$$

Esta métrica penaliza erros maiores de forma mais acentuada devido à operação de quadrado, tornando-a particularmente sensível a *outliers* (Willmott & Matsuura 2005). No contexto de PINNs, o MSE tem sido amplamente utilizado para avaliar a precisão das soluções numéricas em comparação com soluções analíticas ou experimentais (Raissi et al. 2019).

Raiz do Erro Quadrático Médio (RMSE)

O RMSE é simplesmente a raiz quadrada do MSE, calculado como:

$$\text{RMSE}_j = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_{ij} - \hat{y}_{ij})^2} \quad (4.42)$$

Esta métrica mantém as mesmas propriedades do MSE, mas apresenta a vantagem de estar na mesma unidade das variáveis originais, facilitando a interpretação (Chai & Draxler 2014).

4.3.2 Implementação das Métricas de Avaliação

A implementação das métricas de avaliação foi realizada utilizando o *framework* PyTorch, que oferece suporte nativo a operações tensoriais e facilita o cálculo eficiente em lotes de dados (Paszke et al. 2019). A função `score` foi desenvolvida para calcular simultaneamente todas as cinco métricas para cada uma das quatro variáveis de saída, retornando os resultados em um formato tabular através da biblioteca Pandas (Wes McKinney 2010).

Esta abordagem modular e vetorizada permite a avaliação eficiente do modelo em conjuntos de dados de teste potencialmente grandes, além de facilitar a comparação entre diferentes configurações do modelo ou entre o modelo PIKAN e outras abordagens. A implementação segue as melhores práticas de programação científica, com funções bem documentadas e interfaces consistentes (Wilson et al. 2014).

4.3.3 Justificativa da Escolha das Métricas

A escolha das cinco métricas complementares para avaliação do modelo PIKAN foi fundamentada em considerações teóricas e práticas relevantes para o problema específico do traçamento de raios sísmicos:

- O R^2 foi selecionado por sua capacidade de quantificar a proporção da variância explicada pelo modelo, oferecendo uma medida normalizada que facilita a interpretação e comparação entre diferentes variáveis (Draper & Smith 1966).
- O MAE foi escolhido por sua robustez a *outliers* e interpretabilidade direta na mesma unidade das variáveis originais, sendo particularmente útil para avaliar o erro médio esperado nas coordenadas espaciais (Chai & Draxler 2014).
- O MAPE foi incluído para fornecer uma medida relativa do erro, permitindo comparações entre variáveis com diferentes escalas e ordens de magnitude (Hyndman & Koehler 2006).
- O MSE foi selecionado por sua sensibilidade a erros maiores, o que é importante em aplicações físicas onde desvios significativos podem indicar violações das leis físicas subjacentes (Willmott & Matsuura 2005).
- O RMSE foi incluído por combinar a sensibilidade do MSE a erros maiores com a vantagem de estar na mesma unidade das variáveis originais, facilitando a interpretação (Chai & Draxler 2014).

A utilização conjunta destas métricas proporciona uma visão abrangente do desempenho do modelo, capturando diferentes aspectos da qualidade preditiva e permitindo uma avaliação mais robusta e informativa (Botchkarev 2019).

4.3.4 Avaliação Qualitativa

Além da avaliação quantitativa baseada em métricas, foi realizada uma análise qualitativa do desempenho do modelo através da comparação visual entre os raios sísmicos obtidos pelo modelo PIKAN e aqueles obtidos analiticamente via método de Runge-Kutta de segunda ordem. Esta abordagem complementar é fundamental em problemas físicos, onde a conformidade com princípios físicos subjacentes é tão importante quanto a precisão numérica (Karniadakis et al. 2021).

A avaliação qualitativa foi estruturada em três aspectos principais:

1. **Conformidade com as leis físicas:** Verificação visual da aderência dos raios previstos às equações canônicas de Hamilton que governam a propagação de ondas sísmicas (Červený 2001).
2. **Comportamento em regiões de elevado gradiente de velocidade:** Análise da capacidade do modelo em capturar corretamente a curvatura dos raios em regiões com variações significativas no campo de velocidade (Rawlinson et al. 2008).
3. **Generalização para condições iniciais não vistas:** Avaliação da robustez do modelo quando aplicado a condições iniciais (posição e ângulo) não presentes no conjunto de treinamento (Liu et al. 2025).

Para todos os aspectos, foram gerados gráficos comparativos mostrando os raios previstos pelo modelo PIKAN e os raios de referência calculados pelo método de Runge-Kutta. Esta visualização permite identificar padrões sistemáticos de erro e avaliar qualitativamente a capacidade do modelo em generalizar o conhecimento físico aprendido.

4.3.5 Protocolo de Avaliação

O protocolo de avaliação seguiu uma metodologia rigorosa para garantir a validade e reprodutibilidade dos resultados, onde ambos os dados, códigos e análises utilizados neste trabalho estão disponibilizados em um repositório no GitHub (Pineau et al. 2021). Assim, os seguintes passos foram executados:

1. **Separação estrita dos dados:** O conjunto de teste foi mantido completamente isolado durante todas as fases de desenvolvimento do modelo, incluindo treinamento, validação e ajuste de hiperparâmetros, para evitar vazamento de informações (*data leakage*) (Kaufman et al. 2012).
2. **Avaliação em múltiplas condições:** O modelo foi avaliado em diferentes configurações de campo de velocidade e condições iniciais, para testar sua robustez e capacidade de generalização (Shukla et al. 2024).
3. **Cálculo das métricas:** As cinco métricas quantitativas (R^2 , MAPE, MAE, MSE, RMSE) foram calculadas para cada uma das quatro variáveis de saída, resultando em uma matriz de avaliação completa.
4. **Visualização dos resultados:** Foram gerados gráficos comparativos entre os raios previstos e os raios de referência, permitindo uma avaliação qualitativa do desempenho do modelo.

Este protocolo sistemático permite uma avaliação abrangente do modelo PIKAN, considerando tanto aspectos quantitativos quanto qualitativos do seu desempenho.

4.3.6 Considerações sobre Avaliação em Modelos Híbridos PIKAN

A avaliação de modelos híbridos que combinam redes neurais com informação física (PINNs) e redes de Kolmogorov-Arnold (KANs) apresenta desafios específicos que foram considerados na metodologia adotada. Conforme destacado por (Zhang et al. 2025), estes modelos devem ser avaliados não apenas em termos de precisão preditiva, mas também quanto à sua capacidade de respeitar as restrições físicas subjacentes.

No contexto do PIKAN, isso implica verificar se os raios previstos satisfazem as equações canônicas de Hamilton para o problema de traçamento de raios sísmicos. Esta verificação foi realizada tanto quantitativamente, através do cálculo do resíduo físico nas equações diferenciais, quanto qualitativamente, através da análise visual das trajetórias previstas, tendo como objetivo principal avaliar a capacidade do modelo em generalizar para condições não vistas durante o treinamento, um aspecto crítico para aplicações práticas em geofísica exploratória (Wang et al. 2025).

Capítulo 5

Experimentos e Resultados

Neste capítulo são apresentados os experimentos realizados nesta pesquisa, bem como os respectivos resultados obtidos.

Em consonância com o protocolo de avaliação descrito na Subseção 4.3.5 deste trabalho, foram realizados dois experimentos principais. O primeiro envolve a utilização de um campo de velocidades simples e parametrizado, definido por uma equação linear. O segundo, por sua vez, explora um modelo não linear e mais complexo, caracterizado por descontinuidades e cuja definição é baseada em uma nuvem de pontos, ao invés de uma equação linear.

Para ambos os experimentos, as posições (x, z) são expressas na escala de quilômetros, as velocidades $v(x, z)$ são medidas em quilômetros por segundo, e o vetor de vagarosidade (p_x, p_z) é representado em unidades de segundos por quilômetro. Essas escalas foram escolhidas de forma a assegurar que as variáveis não apresentem diferenças significativas em suas ordens de grandeza.

Para facilitar a compreensão e análise dos resultados, o capítulo será estruturado em duas seções, sendo que cada uma delas abordará detalhadamente o experimento correspondente e os resultados obtidos.

5.1 Modelo Parametrizado

Para os experimentos iniciais com o modelo PIKAN, foi utilizado um campo de velocidade parametrizado, definido de forma analítica pela seguinte equação:

$$v(x, z) = 1.5 + 0.3x + z \quad (5.1)$$

Esta formulação representa um modelo de velocidade com gradiente linear tanto na direção horizontal quanto vertical, onde a velocidade aumenta com a profundidade (z) e com a distância horizontal (x). A escolha deste modelo paramétrico simplificado permite validar o comportamento do método proposto em um cenário onde as trajetórias dos raios podem ser calculadas com alta precisão através de métodos numéricos convencionais, servindo assim como referência para avaliação do desempenho do modelo PIKAN.

5.1.1 Geração dos Dados Sintéticos

A geração dos dados sintéticos para treinamento, validação e teste do modelo, seguiu o Algoritmo 1, explicado na Subseção 4.1.2 deste trabalho. A implementação foi feita em Python 3.10.15, com ajuda das bibliotecas Pandas, Numpy e Matplotlib, através da classe `DataGenerator`, que encapsula todo o fluxo de trabalho necessário para produzir trajetórias de raios sísmicos no campo de velocidade parametrizado. O processo completo de geração de dados segue as etapas descritas a seguir.

Definição do Domínio Computacional

O primeiro passo consiste em definir os limites do domínio computacional para as variáveis x e z , de modo que:

$$x_{\min}, x_{\max}, z_{\min}, z_{\max} \in \mathbb{R}^+, \quad x_{\max} \geq x_{\min} + dg, \quad z_{\max} \geq z_{\min} + dg \quad (5.2)$$

Para os experimentos realizados, foram adotados os seguintes limites:

$$x_{\text{range}} = (0, 3.6) \quad (5.3)$$

$$z_{\text{range}} = (0, 1.8) \quad (5.4)$$

Estes valores definem uma região retangular que representa o meio de propagação das ondas sísmicas, com dimensões adequadas para capturar a curvatura dos raios induzida pelo gradiente de velocidade.

Discretização do Campo de Velocidade

O campo de velocidade contínuo é discretizado em uma grade regular com espaçamento dg_{over} entre pontos adjacentes. Para garantir uma representação precisa do campo, foi utilizado um valor pequeno para este parâmetro:

$$dg_{\text{over}} = 1 \times 10^{-2} \quad (5.5)$$

A partir desta discretização fina, são gerados vetores de coordenadas para os eixos x e z :

$$\mathbf{x}_{\text{over}} = \{x_{\min}, x_{\min} + dg_{\text{over}}, \dots, x_{\max}\} \quad (5.6)$$

$$\mathbf{z}_{\text{over}} = \{z_{\min}, z_{\min} + dg_{\text{over}}, \dots, z_{\max}\} \quad (5.7)$$

O campo de velocidade é então calculado para cada ponto da grade, aplicando a equação paramétrica:

$$\mathbf{v}_{\text{depth}}(i, j) = 1.5 + 0.3 \cdot \mathbf{x}_{\text{over}}(j) + \mathbf{z}_{\text{over}}(i) \quad (5.8)$$

Subamostragem e Interpolação B-splines

Como explicado na Subseção 4.1.1, o campo de velocidade discretizado é subamostrado por um fator factor :

$$s = 20 \quad (5.9)$$

Isto resulta em uma grade mais esparsa com espaçamento efetivo:

$$dg = s \cdot dg_{\text{over}} = 0.2 \quad (5.10)$$

O campo de velocidade subamostrado é então utilizado para calcular os coeficientes da interpolação B-splines bidimensional, que permite reconstruir o campo discretizado sem descontinuidades a partir dos valores discretos:

$$\mathbf{c}[i, j] = \text{coeffs_bsplines_2d}(\mathbf{v}[i, j]) \quad (5.11)$$

Definição das Condições Iniciais

Para cada raio sísmico a ser traçado, são definidas as seguintes condições iniciais:

1. **Posição inicial:** (x_0, z_0) dentro do domínio computacional.
2. **Ângulo inicial:** θ_0 em graus, medido a partir da vertical no sentido anti-horário.

Para os experimentos realizados, foram utilizados os seguintes intervalos para as condições iniciais:

$$\theta_{\text{range}} = (45^\circ, 75^\circ) \quad (5.12)$$

$$x_{0\text{range}} = (0.25, 2.25) \quad (5.13)$$

$$z_{0\text{range}} = (1.0, 1.75) \quad (5.14)$$

O ângulo inicial $\theta_0 \in \theta_{\text{range}}$ é convertido para radianos e ajustado para o sistema de coordenadas utilizado:

$$\theta_{0_p} = \pi \cdot \frac{180^\circ - \theta_0}{180^\circ} \quad (5.15)$$

Os coeficientes são então espelhados e a velocidade no ponto inicial é calculada através da interpolação B-splines:

$$v_0 = \text{interp2d_bsplines_core}(\mathbf{c}^{\text{espelhado}}, \frac{z_0}{dg}, \frac{x_0}{dg}), \quad (5.16)$$

onde $\mathbf{c}^{\text{espelhado}}$ são os coeficientes B-splines com tratamento de borda por espelhamento.

O vetor de vagarosidade inicial é então calculado a partir do ângulo e da velocidade:

$$\begin{cases} p_{x_0} = \frac{\sin \theta_{0_p}}{v_{0_p}} \\ p_{z_0} = \frac{\cos \theta_{0_p}}{v_{0_p}} \end{cases} \quad (5.17)$$

O estado inicial completo do raio é representado pelo vetor:

$$\mathbf{r}_0 = [x_0, z_0, p_{x_0}, p_{z_0}] \quad (5.18)$$

Integração Numérica das Trajetórias

Com as condições iniciais e o campo de velocidade definidos, a trajetória do raio é calculada através do método de Runge-Kutta de segunda ordem, conforme descrito no Capítulo 4. Os parâmetros de integração utilizados foram:

$$dt = 1 \times 10^{-3} \text{ s} \quad (5.19)$$

$$t_{\max} = 0.4 \text{ s} \quad (5.20)$$

O algoritmo de traçamento de raios retorna três conjuntos de dados:

1. `tray`: Vetor de tempos discretos $t \in [0, t_{\max}]$;
2. `rt`: Matriz de estados $[x(t), z(t), p_x(t), p_z(t)]$ para cada instante de tempo discreto;
3. `dray`: Matriz de derivadas $[\dot{x}(t), \dot{z}(t), \dot{p}_x(t), \dot{p}_z(t)]$ para cada instante de tempo discreto.

Organização dos Dados

Os resultados do traçamento de raios são organizados em um *DataFrame* do Pandas, contendo as seguintes colunas:

- `x`, `z`: Coordenadas espaciais do raio em cada instante de tempo;
- `px`, `pz`: Componentes do vetor de vagarosidade em cada instante de tempo;
- `x0`, `z0`: Posição inicial do raio;
- `theta0_p`: Ângulo inicial em radianos;
- `t`: Tempo de trânsito desde a origem;
- `dg`: Espaçamento da grade;
- `v`: Valor da velocidade no ponto;
- `dxdt`: Derivada da posição x pelo tempo t ;
- `dzdt`: Derivada da posição z pelo tempo t ;
- `dpxdt`: Derivada da componente x do vetor de vagarosidade $\mathbf{p}$ pelo tempo t ;
- `dpzdt`: Derivada da componente z do vetor de vagarosidade $\mathbf{p}$ pelo tempo t ;

Esta estrutura de dados facilita o posterior processamento e a alimentação do modelo PIKAN durante as fases de treinamento, validação e teste.

5.1.2 Geração do Conjunto de Dados Completo

Para gerar um conjunto de dados abrangente, o processo descrito anteriormente foi repetido para múltiplas combinações de condições iniciais, variando sistematicamente os valores de x_0 , z_0 e θ_0 dentro dos intervalos especificados. Especificamente, foram utilizados os seguintes passos:

1. Discretização dos intervalos de condições iniciais com espaçamento de 0.25 km para x_0 e z_0 , e 5° para θ_0 .
2. Execução do traçamento de raios para cada combinação de condições iniciais.
3. Concatenação dos *DataFrames* resultantes em um único conjunto de dados.

O conjunto de dados final contém 101.052 pontos, correspondendo a 252 trajetórias distintas com diferentes condições iniciais. Este volume de dados foi considerado adequado para capturar a diversidade de comportamentos dos raios sísmicos no campo de velocidade parametrizado e permitir um treinamento efetivo do modelo PIKAN.

Em seguida, como descrito na Seção 4.1, o conjunto de dados completo foi dividido em três subconjuntos para as fases de treinamento, validação e teste, seguindo a proporção padrão de 60%, 20% e 20%, respectivamente. Importante reforçar que a divisão foi realizada ao nível das trajetórias completas, e não dos pontos individuais, para evitar vazamento de informações entre os conjuntos. Isto significa que todos os pontos pertencentes a uma mesma trajetória foram alocados exclusivamente a um dos três subconjuntos.

Após isso, os pesos da função física foram computados para o conjunto de treinamento, assim como descrito na Subseção 4.1.3 deste trabalho. Por fim, o conjunto de dados de treinamento final seguiu o formato¹ da Tabela 5.1.

x_0	z_0	θ_0	t	π_weight	$dxdt$	$dzdt$	$dpxdt$	$dpzdt$	x	z	px	pz
0.25	1.00	2.36	0.00	10.00	1.82	-1.82	-0.12	-0.39	0.25	1.00	0.27	-0.27
0.25	1.00	2.36	0.00	4.54	1.82	-1.82	-0.12	-0.39	0.25	1.00	0.27	-0.27
0.25	1.00	2.36	0.00	4.54	1.82	-1.83	-0.12	-0.39	0.25	1.00	0.27	-0.28
0.25	1.00	2.36	0.00	4.54	1.82	-1.83	-0.12	-0.39	0.26	0.99	0.27	-0.28
0.25	1.00	2.36	0.00	4.54	1.81	-1.83	-0.12	-0.39	0.26	0.99	0.27	-0.28

Tabela 5.1: Exemplo das cinco primeiras amostras do conjunto de dados de treinamento para o modelo de velocidade parametrizado.

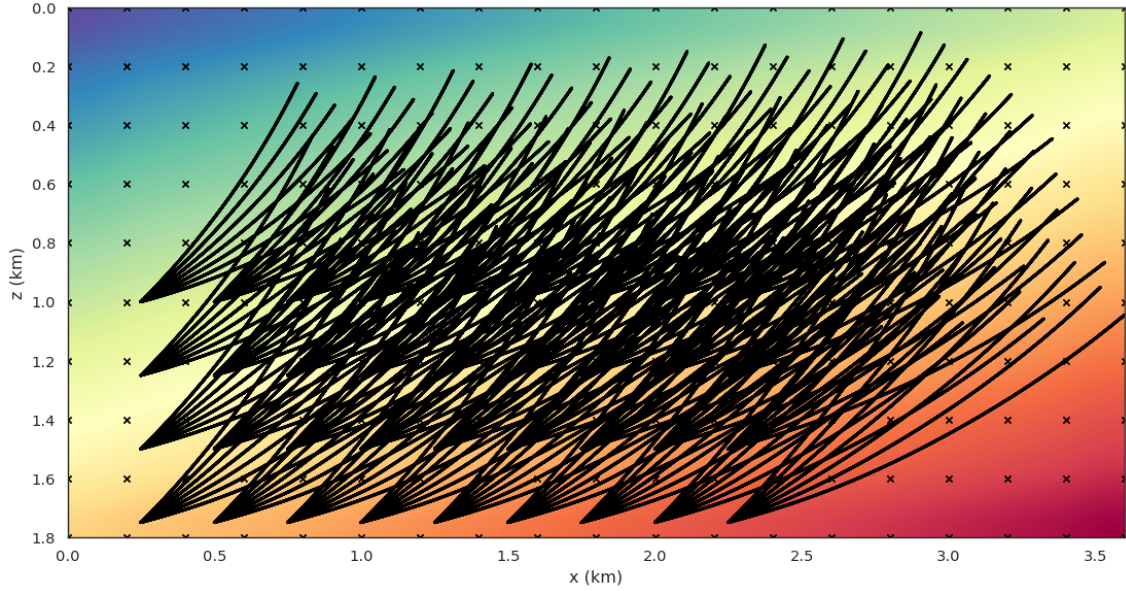
5.1.3 Visualização dos Dados Gerados

Para validar a qualidade dos dados gerados e obter *insights* sobre o comportamento dos raios sísmicos no campo de velocidade parametrizado, foram produzidas visualizações das trajetórias e do campo de velocidade. A Figura 5.1 ilustra o conjunto total das trajetórias geradas, sobrepostas ao campo de velocidade.

Estas visualizações confirmam que as trajetórias dos raios exibem a curvatura esperada devido ao gradiente de velocidade, com os raios tendendo a se curvar em direção às regiões de menor velocidade, conforme previsto pela teoria do traçamento de raios sísmicos.

¹Os valores foram arredondados para duas casas decimais para melhorar a visualização da tabela.

Figura 5.1: Raios traçados para o modelo de velocidade parametrizado.



Fonte: Elaborado pelo autor (2025).

5.1.4 Refinamento de Hiperparâmetro: ajuste do tamanho da grade na KAN

Um aspecto crítico no desenvolvimento do modelo PIKAN é a determinação do tamanho ideal da grade utilizada na Rede de Kolmogorov-Arnold (KAN). Este hiperparâmetro controla diretamente a capacidade de representação do modelo, influenciando tanto a performance preditiva, quanto a eficiência computacional do modelo. Esta subseção descreve o procedimento experimental adotado para determinar o tamanho ótimo da grade, bem como os resultados obtidos e suas implicações para o desempenho do modelo.

Metodologia de Ajuste do Hiperparâmetro

Para determinar o tamanho ideal da grade na arquitetura KAN, foi implementado um procedimento de busca de hiperparâmetros, avaliando diferentes configurações e analisando seu impacto nas métricas de desempenho. Especificamente, foram testados sete tamanhos de grade diferentes:

$$\mathcal{G} \in \{4, 8, 12, 16, 20, 24\} \quad (5.21)$$

A arquitetura do modelo PIKAN foi configurada com a seguinte estrutura:

- **Camadas de entrada e saída:** Dimensionadas de acordo com o número de características de entrada e variáveis alvo, respectivamente. Ou seja, 4 de entrada e 4 de saída.

- **Camadas intermediárias:** Duas camadas intermediárias com configurações [4, 2] e [3, 1], representando o número de nós somadores e o número de nós multiplicadores, respectivamente (Liu et al. 2025).
- **Grau polinomial:** $k = 3$, correspondendo a B-splines cúbicas.
- **Intervalo da grade:** $[-5, 5]$, definindo o domínio de operação das funções splines.

O treinamento foi realizado utilizando o otimizador Adam com taxa de aprendizado de $\eta = 10^{-3}$ e função de perda de dados dada pelo Erro Médio Quadrático (MSE, do inglês *Mean Squared Error*).

A função de perda física utilizada é definida pela Equação 4.16. As derivadas $\frac{dx_i}{dt}$ e $\frac{dp_i}{dt}$ foram obtidas a partir da diferenciação automática do PyTorch, a cada passo do treinamento. Enquanto $\nabla v(\mathbf{x}_i)$ foram obtidas previamente, durante a criação do conjunto de dados.

Durante o treinamento da rede, percebeu-se a dificuldade da PIKAN em representar os pontos iniciais (para $t = 0$) corretamente, o que resultava no traçamento de raio muito diferente do original, obtido pelo RK2. Para superar esta dificuldade, um erro quadrático médio foi somado a função de perda física, aumentando a importância dos pontos iniciais na predição do modelo.

Os termos de regularização (entropia, L1, coeficientes e diferenças de coeficientes) foram desativados neste experimento específico, permitindo uma avaliação mais direta do impacto do tamanho da grade na capacidade de representação do modelo.

Protocolo Experimental

O protocolo experimental seguiu as seguintes etapas:

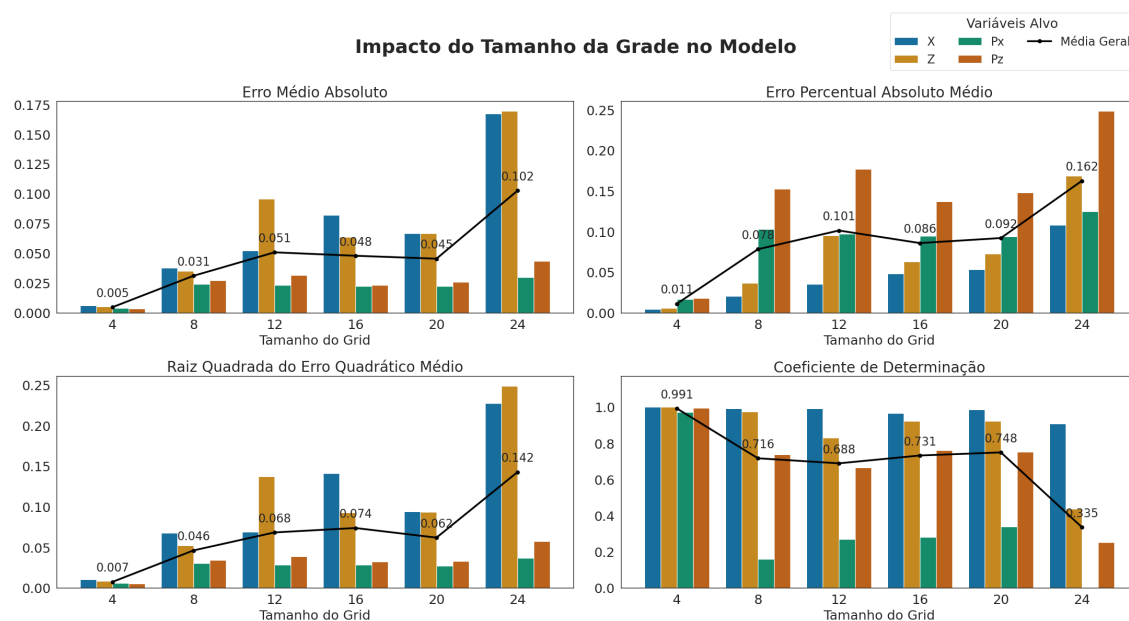
1. **Inicialização:** O modelo foi inicializado com o menor tamanho de grade (4) e treinado por um número fixo de épocas (`save_steps = 20`).
2. **Refinamento progressivo:** Após o treinamento inicial, o modelo foi refinado sequencialmente para cada tamanho de grade subsequente na lista $\mathcal{G}$, utilizando o método `refine` da classe KAN. Este método preserva o conhecimento adquirido enquanto adapta a estrutura para acomodar uma grade mais densa.
3. **Treinamento por estágios:** Para cada tamanho de grade, o modelo foi treinado por `save_steps` épocas adicionais, com atualizações adaptativas da grade durante o treinamento. Estas atualizações foram controladas pelos parâmetros:
 - `update_grid = True`: Habilitando o mecanismo de atualização adaptativa.
 - `stop_grid_update_step = save_steps`: Limitando as atualizações ao número total de épocas para cada tamanho de grade.
 - `n_grid_update = 10`: Definindo o número total de atualizações a serem realizadas.
4. **Avaliação:** Após o treinamento com cada tamanho de grade, o modelo foi avaliado no conjunto de validação, calculando as métricas de desempenho (R^2 , MAE, MAPE, MSE, RMSE) para cada variável alvo (x , z , p_x , p_z).
5. **Armazenamento de resultados:** Os resultados da avaliação e o estado do modelo foram salvos para cada configuração, permitindo análises comparativas posteriores.

Este protocolo resultou em um total de 140 épocas de treinamento (20 épocas $\times$ 7 tamanhos de grade), proporcionando uma exploração ampla do espaço de hiperparâmetros relacionado ao tamanho da grade.

Resultados e Análise

A Figura 5.2 apresenta os resultados da avaliação do modelo para diferentes tamanhos de grade, considerando quatro métricas principais: Erro Absoluto Médio (MAE), Erro Percentual Absoluto Médio (MAPE), Raiz do Erro Quadrático Médio (RMSE) e Coeficiente de Determinação (R^2). Cada métrica é apresentada para as quatro variáveis alvo (x , z , p_x , p_z), permitindo uma análise detalhada do desempenho do modelo.

Figura 5.2: Impacto do tamanho da grade nas métricas de avaliação do modelo PIKAN para o modelo de velocidade parametrizado. As barras coloridas representam as métricas para cada variável alvo, enquanto a linha preta indica a média do modelo.



Fonte: Elaborado pelo autor (2025).

A análise dos resultados revela padrões significativos:

1. **Erro Absoluto Médio (MAE):** Observa-se um aumento gradual do erro com o aumento do tamanho da grade, com um salto notável para o tamanho 24. O menor MAE médio (0.005) é obtido com o tamanho 4, enquanto o maior (0.102) ocorre com o tamanho 24.
2. **Erro Percentual Absoluto Médio (MAPE):** Segue um padrão similar ao MAE, com valores crescentes à medida que o tamanho da grade aumenta. O MAPE médio varia de 0.011 (tamanho 4) a 0.162 (tamanho 24).

3. **Raiz do Erro Quadrático Médio (RMSE):** Apresenta comportamento consistente com as métricas anteriores, com valores crescentes para tamanhos de grade maiores. O RMSE médio varia de 0.007 (tamanho 4) a 0.142 (tamanho 24).
4. **Coefficiente de Determinação (R^2):** Exibe um padrão inverso, com valores decrescentes à medida que o tamanho da grade aumenta. O R^2 médio varia de 0.991 (tamanho 4) a 0.335 (tamanho 24).

Um aspecto notável é a diferença de comportamento entre as variáveis alvo. As coordenadas espaciais (x e z) geralmente apresentam erros normalizados menores em comparação com as componentes do vetor de vagarosidade (p_x e p_z), especialmente para tamanhos de grade maiores. Isto sugere que a predição do vetor de vagarosidade é mais sensível ao aumento do tamanho da grade da posição, possivelmente devido à sua menor ordem de grandeza em relação à posição, o que amplifica o impacto relativo dos erros nas métricas normalizadas. Em outras palavras, como os valores típicos de p_x e p_z são menores em módulo do que os de x e z , uma mesma variação absoluta no erro leva a um erro relativo mais alto para o vetor de vagarosidade.

Para este experimento, os resultados experimentais revelaram uma tendência contrainintuitiva: o desempenho do modelo piora com o aumento do tamanho da grade. Este comportamento pode ser atribuído a vários fatores:

1. **Sobreajuste:** Grades maiores proporcionam maior capacidade de representação, potencialmente levando ao sobreajuste dos dados de treinamento e prejudicando a generalização para dados não vistos;
2. **Desafios de otimização:** O aumento do número de parâmetros associado a grades maiores torna o problema de otimização mais complexo, dificultando a convergência para mínimos globais;
3. **Interação com a função de perda física:** A combinação de grades mais densas com restrições físicas tende a produzir uma paisagem de otimização mais complexa e potencialmente mais acidentada. Isso ocorre porque o aumento da resolução espacial e a imposição de múltiplas restrições reduzem o espaço de soluções viáveis, podendo gerar gradientes mais instáveis e múltiplos mínimos locais durante o treinamento da rede.

Com base nos resultados quantitativos e nas considerações acima, o tamanho de grade 4 emerge como a escolha ótima para o modelo neste experimento com campo de velocidade parametrizado. Esta configuração apresenta o melhor desempenho em todas as métricas avaliadas, com MAE médio de 0.005, MAPE médio de 0.011, RMSE médio de 0.007 e R^2 médio de 0.991.

É importante ressaltar que esta conclusão é específica para o problema e conjunto de dados em questão. Em cenários mais complexos, com campos de velocidade heterogêneos ou descontinuidades acentuadas, tamanhos de grade maiores podem ser necessários para capturar adequadamente as variações do meio, como será discutido na Seção 5.2. Adicionalmente, estratégias alternativas de regularização e otimização poderiam potencialmente mitigar os problemas observados com grades maiores.

5.1.5 Treinamento e Avaliação do Melhor Modelo

Após a análise sistemática do impacto do tamanho da grade na capacidade de representação e no desempenho preditivo do modelo PIKAN, foi selecionado o tamanho de grade $G = 4$ como configuração ótima para o problema de traçamento de raios sísmicos no campo de velocidade parametrizado. Esta subseção apresenta o processo de treinamento refinado do modelo final – com os melhores hiperparâmetros – e os resultados obtidos, tanto em termos quantitativos quanto qualitativos.

Configuração do Treinamento Final

Para o treinamento do modelo final, foi mantida a arquitetura PIKAN com as mesmas especificações estruturais descritas anteriormente, porém com ajustes nos hiperparâmetros de otimização e no protocolo de treinamento. As principais configurações adotadas foram:

- **Tamanho da grade:** $G = 4$, conforme determinado pelo estudo de hiperparâmetros.
- **Número de épocas:** 35, proporcionando tempo suficiente para convergência.
- **Taxa de aprendizagem inicial:** $\eta_0 = 10^{-2}$, uma ordem de magnitude maior que a utilizada no estudo de hiperparâmetros, permitindo exploração mais ampla do espaço de parâmetros nas fases iniciais do treinamento.
- **Agendamento adaptativo da taxa de aprendizagem:** Implementado através do algoritmo `ReduceLRonPlateau`, fornecido pelo PyTorch, com uma redução da taxa de aprendizagem por um fator de 0.1 quando não houver melhoria na perda de validação por 5 épocas consecutivas. Matematicamente, a taxa de aprendizagem na época t é dada por:

$$\eta_t = \begin{cases} \eta_{t-1} \cdot 0.1, & \text{se } \min_{t-5 \leq i < t} \mathcal{L}_{\text{val}}^{(i)} \leq \mathcal{L}_{\text{val}}^{(t)} \\ \eta_{t-1}, & \text{caso contrário} \end{cases} \quad (5.22)$$

- **Atualização da grade:** Realizada a cada 10 épocas, permitindo o refinamento adaptativo da distribuição dos nós da grade em momentos estratégicos do treinamento.

Os demais parâmetros do modelo foram mantidos idênticos aos utilizados no estudo de hiperparâmetros, incluindo a função de perda composta (dados + física), a técnica de *singularity avoiding* e a ponderação adaptativa da perda física.

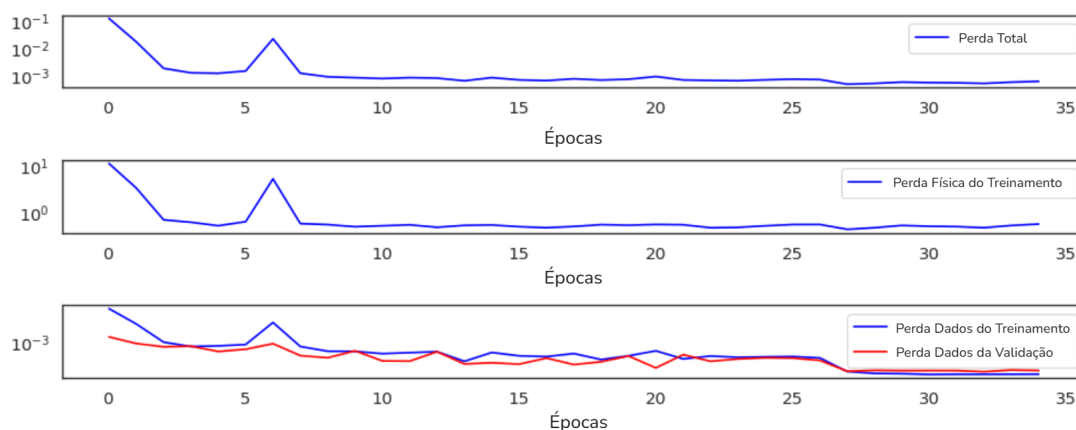
Evolução do Treinamento

A Figura 5.3 apresenta a evolução das diferentes componentes da função de perda ao longo das 35 épocas de treinamento. Os gráficos mostram três aspectos fundamentais do processo de aprendizagem: a perda total de treinamento, a perda física de treinamento e a comparação entre as perdas de dados nos conjuntos de treinamento e validação.

A análise dos gráficos revela padrões significativos no processo de aprendizagem:

1. **Perda total de treinamento:** Observa-se uma rápida diminuição nas primeiras épocas, seguida por um pico próximo à época 7, possivelmente associado à primeira

Figura 5.3: Evolução das componentes da função de perda durante o treinamento do modelo PIKAN no modelo de velocidade parametrizado. Painel superior: perda total de treinamento. Painel central: perda física de treinamento. Painel inferior: comparação entre as perdas de dados nos conjuntos de treinamento e validação.



Fonte: Elaborado pelo autor (2025).

atualização da grade. Após este ponto, a perda estabiliza-se em torno de 10^{-3} , com pequenas flutuações nas épocas subsequentes.

2. **Perda física de treinamento:** Apresenta comportamento similar à perda total, com redução acentuada nas fases iniciais e estabilização após a época 10. O pico na época 7 é mais pronunciado nesta componente, sugerindo que a atualização da grade afeta principalmente a satisfação das restrições físicas.
3. **Perdas de dados:** As curvas de treinamento e validação exibem comportamento consistente ao longo de todo o processo, com valores muito próximos entre si. Esta proximidade é um indicador positivo de que o modelo não está sobreajustado ou subajustado, mantendo capacidade de generalização similar para dados vistos e não vistos durante o treinamento.

Um aspecto notável é a ausência de divergência entre as perdas de treinamento e validação, mesmo após 35 épocas. Este comportamento sugere que a combinação do tamanho de grade compacto ($G = 4$) com as atualizações periódicas da grade e o agendamento adaptativo da taxa de aprendizagem foi eficaz em prevenir o sobreajuste, um problema comum em modelos com alta capacidade de representação.

Avaliação Quantitativa

Após o treinamento, o modelo foi avaliado no conjunto de teste, que contém trajetórias de raios com condições iniciais não vistas durante o treinamento ou validação. A Tabela 5.2 apresenta as métricas de desempenho obtidas para cada uma das quatro variáveis alvo: posição (x , z) e componentes do vetor de vagarosidade (p_x , p_z).

	x	z	p_x	p_z
R^2	0.9998	0.9997	0.9952	0.9991
MAPE	0.36%	0.42%	0.69%	0.86%
MAE	0.0051	0.0038	0.0016	0.0015
RMSE	0.004690	0.002431	0.001778	0.003084

Tabela 5.2: Métricas de avaliação para as predições do PIKAN no traçamento de raios sísmicos.

Os resultados quantitativos demonstram um excelente desempenho do modelo PIKAN:

- **Coefficiente de Determinação (R^2):** Valores extremamente próximos de 1 para todas as variáveis alvo, indicando que o modelo explica praticamente toda a variância nos dados. Mesmo o R^2 mínimo de 0,9952 para p_x ainda é uma métrica excelente.
- **Erro Percentual Absoluto Médio (MAPE):** Valores inferiores a 1% para todas as variáveis, com o menor erro (0,36%) para a coordenada x e o maior (0,86%) para a componente p_z do vetor de vagarosidade.
- **Erro Absoluto Médio (MAE):** Valores na ordem de 10^{-3} para as coordenadas espaciais, representando um erro médio de alguns metros, e 10^{-4} para as componentes do vetor de vagarosidade, demonstrando alta precisão nas predições.
- **Raiz do Erro Quadrático Médio (RMSE):** Valores consistentemente baixos, com o menor erro (0,001778) para p_x e o maior (0,004690) para x .

É notável que, apesar da simplicidade da arquitetura (tamanho de grade $G = 4$), o modelo conseguiu capturar com alta precisão tanto as posições quanto as componentes do vetor de vagarosidade ao longo das trajetórias dos raios. Este resultado valida a escolha do tamanho de grade compacto e demonstra a eficácia da abordagem PIKAN para o problema do traçamento de raios sísmicos.

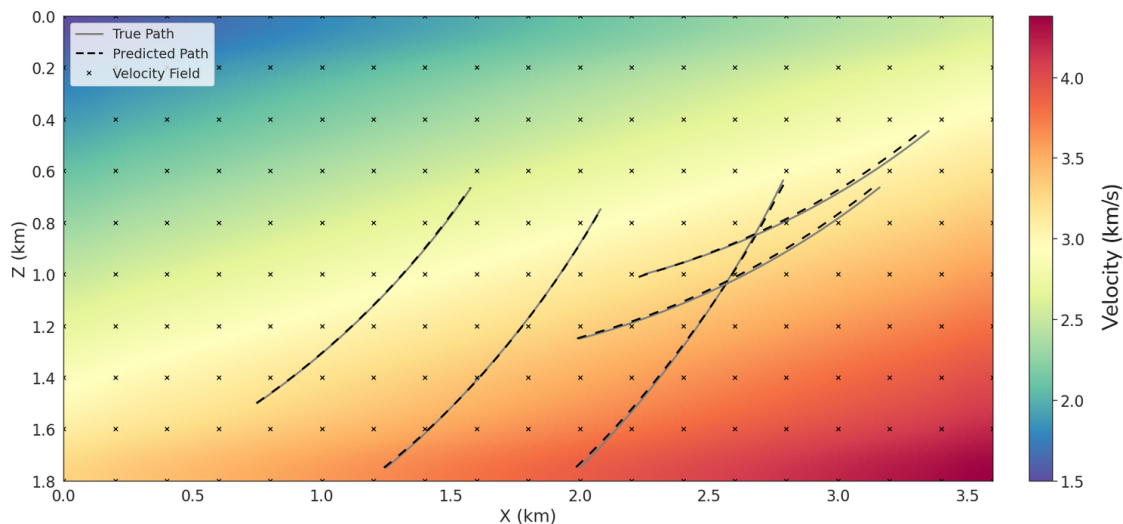
Avaliação Qualitativa

Além da avaliação quantitativa, foi realizada uma inspeção visual das trajetórias preditas pelo modelo PIKAN em comparação com as trajetórias de referência calculadas pelo método de Runge-Kutta de segunda ordem. A Figura 5.4 apresenta esta comparação para um subconjunto representativo de raios no campo de velocidade parametrizado.

A inspeção visual confirma o excelente desempenho do modelo PIKAN, com as trajetórias preditas muito próximas das trajetórias de referência. Observa-se que:

1. O modelo preserva a física subjacente ao fenômeno, respeitando as equações canônicas de Hamilton que governam a propagação dos raios sísmicos.
2. O modelo captura corretamente o comportamento dos raios em diferentes ângulos de incidência e posições iniciais, demonstrando boa capacidade de generalização.
3. As trajetórias preditas seguem fielmente a curvatura característica dos raios sísmicos em meios com gradiente de velocidade, com maior curvatura nas regiões de transição mais acentuada.

Figura 5.4: Comparação visual entre trajetórias de raios sísmicos calculadas pelo método de Runge-Kutta de segunda ordem (linhas sólidas cinzas) e previstas pelo modelo PIKAN (linhas tracejadas pretas). O campo de velocidade parametrizado é representado pelo gradiente de cores, com valores variando de 1,5 km/s (azul) a 4,5 km/s (vermelho).



Fonte: Elaborado pelo autor (2025).

4. As pequenas discrepâncias visíveis em alguns pontos são consistentes com os baixos valores de erro reportados na avaliação quantitativa.

Desta forma, o modelo PIKAN passa nos três critérios estabelecidos na Subseção 4.3.4, demonstrando que o modelo não apenas reproduz os valores numéricos com alta precisão, mas também se ajustou às equações que modelam o fenômeno físico de propagação de onda sísmicas em meios heterogêneos.

5.2 Modelo Não Parametrizado

Embora os resultados obtidos no modelo de velocidades parametrizado tenham sido excepcionais, no contexto da geofísica esse modelo é simples demais para representar as complexidades da propagação de raios sísmicos em meios heterogêneos. Portanto, faz-se necessário a investigação da metodologia proposta em um modelo de velocidade mais próximo da realidade, isto é, heterogêneo, não-parametrizado e com descontinuidades.

Desta forma, essa seção apresenta os experimentos e resultados preliminares obtidos para o traçamento de raios sísmicos no modelo de velocidade do Marmousi, bem como os futuros experimentos a serem feitos para validar corretamente a metodologia proposta para o traçamento de raios sísmicos.

5.2.1 Geração dos Dados Sintéticos: Marmousi

A geração dos dados sintéticos para treinamento, validação e teste do modelo seguiu um processo análogo ao descrito na Subseção 5.1.1, a única mudança consiste na obtenção dos pontos da grade de velocidades. Além disso, uma vez que o modelo já é discretizado, a etapa de Discretização do Campo de Velocidade não é necessária. Por fim, todas as etapas que forem redundantes, isto é, etapas que já foram explicadas detalhadamente na Subseção 5.1.1, serão apenas citadas durante as explicações a seguir.

O Modelo de Velocidade do Marmousi

O modelo de velocidade do Marmousi é um modelo numérico amplamente utilizado em geofísica e na sismica para simular a propagação de ondas sísmicas em um meio heterogêneo. Desenvolvido por investigadores da indústria petrolífera, o modelo Marmousi é caracterizado por uma complexa estrutura geológica, incluindo camadas de diferentes velocidades e profundidades, que refletem uma seção transversal do subsolo realista. Ele é especialmente útil para testar e validar métodos de inversão sísmica e técnicas de imagem de subsuperfície, devido à sua complexidade e à variedade de características geológicas, como saltos de velocidade e variações não-lineares nas camadas. O modelo do Marmousi é considerado um *benchmark* clássico para a avaliação de algoritmos de processamento sísmico e modelagem de ondas em meios heterogêneos.

O modelo de velocidade do Marmousi utilizado neste trabalho pode ser obtido em (Geoazur 2025). O arquivo de interesse com o modelo de velocidade é o `marmousi_vp.bin`, que consiste em um arquivo que armazena dados em formato binário.

Para realizar a leitura dos dados corretamente e converter os dados binários no modelo de velocidade de propagação de ondas P do modelo do Marmousi, é necessário considerar os parâmetros do modelo:

- $n_x = 2301$: número de amostras na dimensão “distância”;
- $n_z = 751$: número de amostras na dimensão “profundidade”;
- $dx = 4$: incremento de distância na dimensão “distância”, isto é, distância entre dois pontos consecutivos no eixo x ;
- $dz = 4$: incremento de distância na dimensão “profundidade”, isto é, distância entre dois pontos consecutivos no eixo z ;

Com esses valores, é possível converter os dados binários para um *array* do Numpy com o tipo de dado *float* de 32 bits (v_p), considerando o formato de armazenamento em *little-endian*. Entretanto, esta operação resulta em um *array* unidimensional de tamanho 1.728.051 com as velocidades em m/s. Assim, para garantir que o modelo estará no formato e grandeza corretos para o modelo PIKAN, o *array* v_p é formatado para (n_x, n_z) , transposto e dividido por 1000.

Definição do Domínio Computacional

Enquanto no modelo parametrizado os limites do domínio computacional para as variáveis x e z foram arbitrados, neste experimento eles seguem o modelo de velocidade do Marmousi, tal que

$$x_{\text{range}} = \left(0, \frac{n_x \cdot dx}{1000}\right) = (0, 9.204) \quad (5.23)$$

$$z_{\text{range}} = \left(0, \frac{n_z \cdot dz}{1000}\right) = (0, 3.004) \quad (5.24)$$

Demais Etapas

As etapas de subamostragem e interpolação B-splines, definição das condições iniciais, integração numérica das trajetórias, e organização dos dados são todas exatamente iguais às etapas detalhadas na Subseção 5.1.1. Os hiperparâmetros utilizados nas etapas citadas estão listados logo abaixo.

$$\begin{aligned} dg_{\text{over}} &= 1 \times 10^{-2} \\ \mathbf{x}_{\text{over}} &= \{x_{\min}, x_{\min} + dg_{\text{over}}, \dots, x_{\max}\} \\ \mathbf{z}_{\text{over}} &= \{z_{\min}, z_{\min} + dg_{\text{over}}, \dots, z_{\max}\} \\ s &= 30 \\ dg &= s \cdot dg_{\text{over}} = 0.3 \\ \theta_{\text{range}} &= (45^\circ, 75^\circ) \\ x_{0\text{range}} &= (4.0, 6.0) \\ z_{0\text{range}} &= (1.0, 2.0) \\ dt &= 1 \times 10^{-3} \text{ s} \\ t_{\text{max}} &= 0.4 \text{ s} \end{aligned}$$

5.2.2 Geração e Visualização do Conjunto de Dados para o Modelo do Marmousi

Assim como descrito na Seção 4.1, o processo foi repetido para múltiplas combinações de condições iniciais, apenas com uma pequena diferença na discretização dos intervalos de condições iniciais, que foi realizado com um espaçamento de 0.1 km para x_0 e z_0 . Os demais passos se mantiveram inalterados.

O conjunto de dados final contém 648.417 pontos, correspondendo a 1617 trajetórias distintas com diferentes condições iniciais. Este volume de dados é quase 6.5x maior que o conjunto de dados utilizado no modelo parametrizado. Esse aumento do conjunto de dados não é por acaso. Uma vez que a complexidade do problema aumentou significativamente, intuitivamente é possível concluir que a complexidade do modelo também deve aumentar para que a PIKAN consiga traçar corretamente os raios sísmicos. Assim, o aumento do conjunto de dados, sobretudo o de treinamento, é uma técnica eficaz e amplamente conhecida para evitar o sobreajuste de modelos mais complexos, desde que a variância dos dados também seja maior (Hastie et al. 2009, Goodfellow et al. 2016).

O conjunto de dados completo foi dividido da mesma forma que no modelo parametrizado: três subconjuntos para as fases de treinamento, validação e teste, seguindo a proporção padrão de 60%, 20% e 20%, respectivamente.

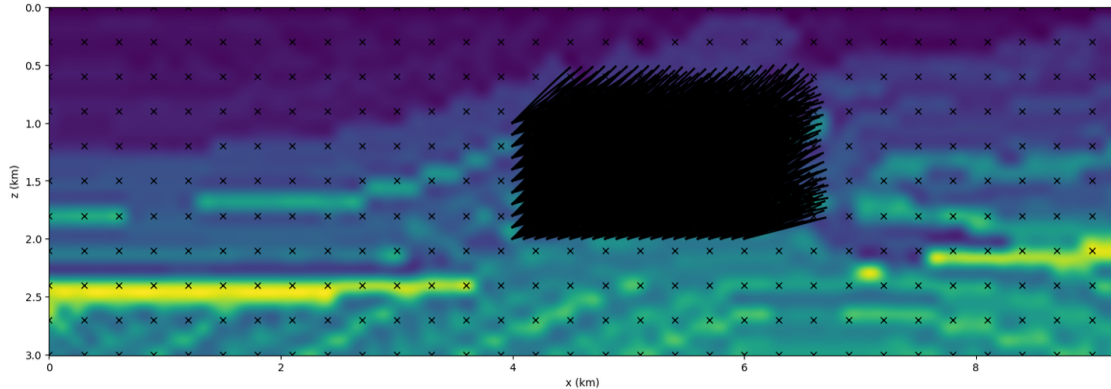
Em seguida, os pesos da função física foram computados para o conjunto de treinamento, tal que o conjunto de dados de treinamento final seguiu o formato² da Tabela 5.3.

x0	z0	theta0_p	t	pi_weight	dxdt	dzdt	dpxdt	dpzdt	x	z	px	pz
4.20	1.90	2.27	0.34	1.27	1.28	-1.22	0.19	-0.05	4.67	1.47	0.41	-0.39
5.90	1.80	2.01	0.33	3.51	1.72	-0.60	-0.00	-0.02	6.47	1.57	0.52	-0.18
5.80	1.90	2.36	0.03	2.29	1.36	-1.34	0.02	0.17	5.84	1.86	0.37	-0.37
5.70	1.10	2.27	0.20	1.96	1.25	-1.11	-0.19	-0.01	5.95	0.88	0.45	-0.40
4.60	1.90	2.01	0.33	1.52	1.49	-0.92	-0.04	-0.22	5.13	1.62	0.49	-0.30

Tabela 5.3: Exemplo de cinco amostras escolhidas aleatoriamente do conjunto de dados de treinamento para o modelo de velocidade do Marmousi.

Por fim, assim como foi feito para o modelo não parametrizado, a Figura 5.5 ilustra o conjunto total das trajetórias geradas, sobrepostas ao campo de velocidade.

Figura 5.5: Raios traçados para o modelo de velocidade do Marmousi a partir do método Runge-Kutta de segunda ordem.



Fonte: Elaborado pelo autor (2025).

5.2.3 Treinamento e Avaliação do Modelo

Refinamento do Fator de Subamostragem

Como discutido na Seção 4.1, o fator de subamostragem (s) constitui um hiperparâmetro central no pré-processamento do campo de velocidade. Ele controla diretamente o grau de suavização introduzido pela interpolação via B-splines: quanto maior o s , mais

²Os valores foram novamente arredondados para duas casas decimais para melhorar a visualização da tabela.

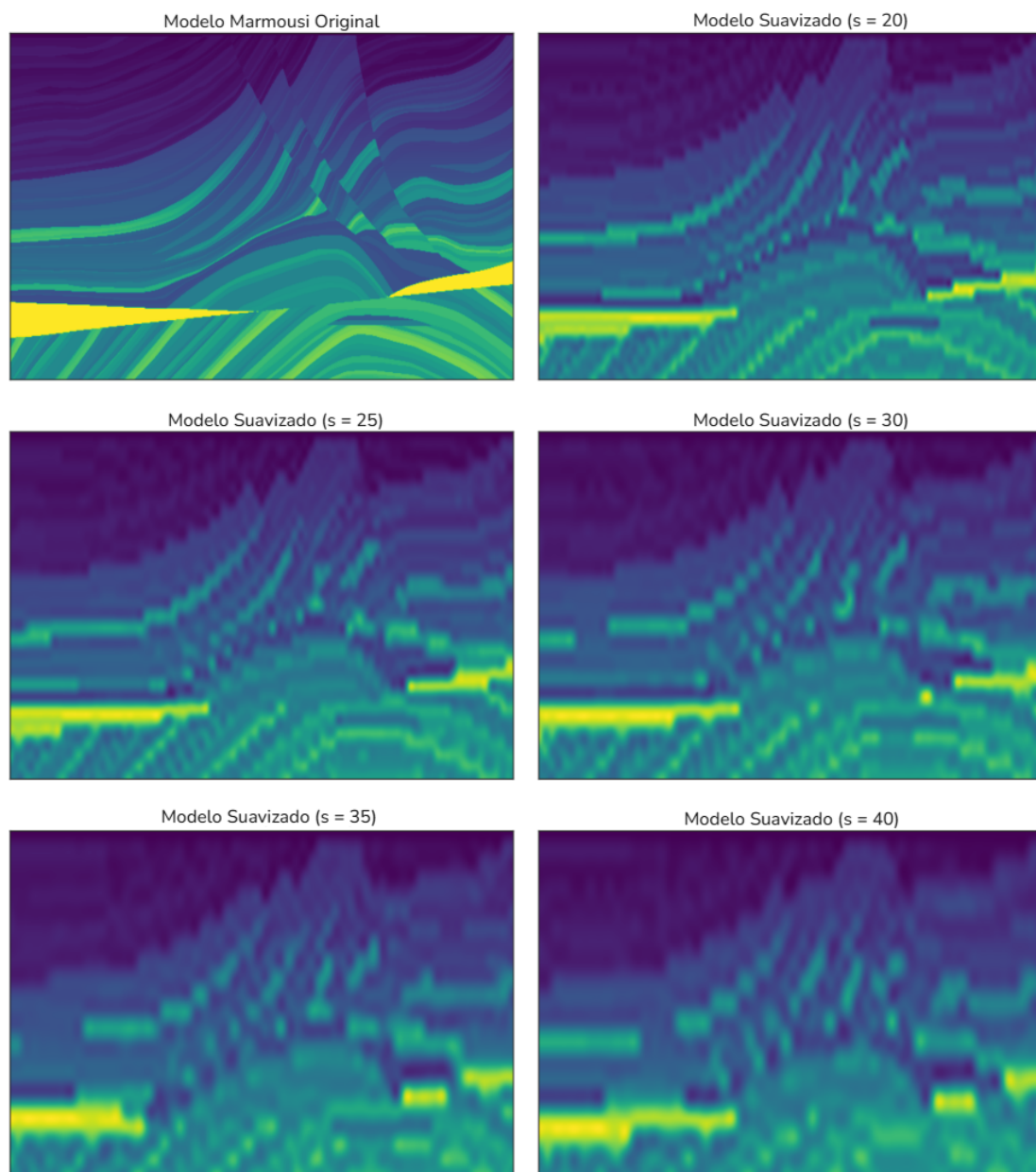
espaçada é a malha de controle e, portanto, mais agressiva a regularização implícita aplicada ao modelo. Essa etapa é particularmente crítica no contexto do traçamento de raios com redes de Kolmogorov–Arnold com informação física (PIKAN), pois o sistema Hamiltoniano que governa a evolução de (X, Z, P_x, P_z) é sensível a gradientes de velocidade. Descontinuidades numéricas ou variações abruptas no campo $v(x, z)$ podem induzir instabilidades no termo de força ∇v^{-1} , degradando a integração e propagando o erro para as componentes do vetor de vagarosidade. Por outro lado, uma suavização excessiva distorce interfaces e heterogeneidades reais, reduzindo a fidelidade física do conjunto de treinamento. Assim, o objetivo ao refinar s é identificar um valor que elimine irregularidades que dificultam a solução do problema direto, sem apagar feições geológicas essenciais do Marmousi. A metodologia completa encontra-se na Subseção 4.2.5.

A análise dos resultados apresentada a seguir evidencia um *trade-off* claro entre tratabilidade numérica e preservação estrutural, corroborado por avaliações qualitativas e quantitativas. A Figura 5.6 compara o Marmousi original com suas versões suavizadas. Para valores baixos de s (e.g., 20 e 25), a interpolação retém maior complexidade do modelo, preservando camadas finas e contrastes laterais, porém ainda permite a permanência de pequenas descontinuidades e variações de alta frequência no campo $v(\mathbf{x})$. Embora sutis no mapa de velocidades, tais irregularidades amplificam o gradiente $\nabla v(\mathbf{x})$ que atua diretamente na evolução de $\mathbf{p}$ pela Equação 2.5, podendo induzir mudanças abruptas na direção do vetor de vagarosidade e, consequentemente, curvaturas não físicas nas trajetórias dos raios. Em contraste, valores altos (e.g., 35 e 40) produzem um campo excessivamente regular, no qual interfaces estratigráficas tornam-se difusas e estruturas de pequena escala, relevantes para os gradientes locais que controlam o traçamento, são atenuadas. Nesse regime, o campo de velocidade passa a representar uma versão filtrada do Marmousi, limitando a capacidade da PIKAN de aprender dinâmicas coerentes com a geologia subjacente.

A Figura 5.7 complementa a inspeção visual ao apresentar métricas de validação em função de s . São mostrados R^2 , MAPE, MAE e RMSE para as quatro variáveis de saída (X, Z, P_x, P_z) ; a curva preta (“Overall”) resume o comportamento médio. Observa-se um platô de desempenho entre $s = 25$ e $s = 35$, com MAPE médio atingindo seus menores valores nessa faixa. Há, contudo, um padrão consistente: o aumento de s tende a elevar o R^2 de X e Z , sugerindo que uma suavização moderada facilita a aprendizagem da geometria global dos raios; entretanto, para P_x e P_z a tendência se inverte em valores altos, indicando perda de informação associada aos gradientes locais que governam a dinâmica do vetor de vagarosidade. Em outras palavras, uma regularização forte melhora a previsibilidade espacial, mas empobrece a física angular/kinemática aprendida pelo modelo.

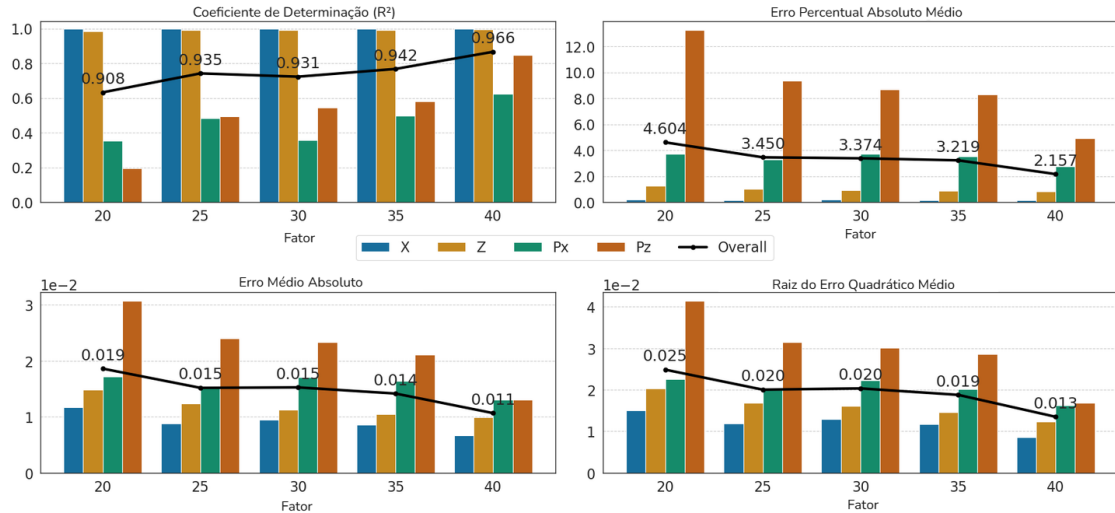
Nota-se que o MAPE “Overall” é minimizado já em $s = 25$ (3,450%), mantendo-se praticamente estável em $s = 30$ (3,374%) e $s = 35$ (3,219%). Essa estabilidade reforça que, dentro desse intervalo, a PIKAN opera próximo ao limite de generalização imposto pelos dados. Ainda assim, as curvas de R^2 e de erros absolutos revelam que valores extremos de s não afetam todas as variáveis de forma homogênea: melhorias marginais em X e Z vêm acompanhadas de degradação sutil porém sistemática em P_x e P_z , evidenciando sensibilidade do modelo ao hiperparâmetro e a necessidade de calibração cuidadosa quando a prioridade é preservar dinâmica física.

Figura 5.6: Comparação visual do Modelo de Velocidade do Marmousi Original (painel superior esquerdo) com as versões suavizadas por interpolação B-splines, variando o fator de subamostragem (s). Valores baixos ($s = 20$ e $s = 25$) preservam contornos e heterogeneidades, mas podem reter discontinuidades locais; $s = 30$ mantém interfaces principais com redução efetiva de variações espúrias; valores altos ($s = 35$ e $s = 40$) promovem suavização excessiva, com perda visível de detalhes finos e contraste entre camadas.



Fonte: Elaborado pelo autor (2025).

Figura 5.7: Avaliação quantitativa do desempenho do modelo PIKAN em função do fator de subamostragem (s) no conjunto de validação.



Fonte: Elaborado pelo autor (2025).

Com base nessa análise, a escolha adotada foi $s = 30$. Esse valor oferece o melhor *trade-off* entre precisão e realismo físico: do ponto de vista quantitativo, mantém o desempenho dentro do platô ótimo; do ponto de vista qualitativo, reduz discontinuidades suficientes para estabilizar o traçamento, preservando simultaneamente as interfaces e heterogeneidades dominantes do Marmousi. Assim, o campo de velocidade utilizado no treinamento da PIKAN permanece representativo da complexidade geológica do problema, mas com suavidade adequada para evitar artefatos numéricos na integração dos raios.

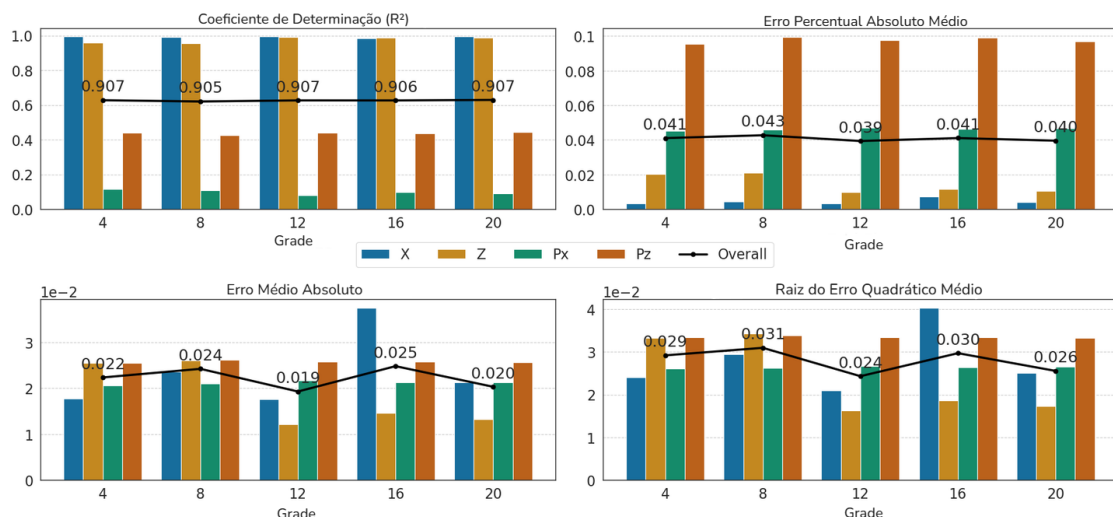
Refinamento do Tamanho de Grade

Conforme descrito na Subseção 4.2.4, o tamanho de grade (G) é um hiperparâmetro intrínseco da arquitetura KAN, pois define a resolução da malha sobre a qual as funções B-splines das camadas são representadas. Em termos práticos, G controla a capacidade de aproximação local da rede: grades mais finas permitem capturar variações mais detalhadas das funções aprendidas, ao custo de maior número de parâmetros e maior tempo de treinamento. Assim, o objetivo deste refinamento foi avaliar a sensibilidade do desempenho da PIKAN à resolução das splines e identificar um valor de G que preservasse a performance preditiva sem impor custos computacionais desnecessários.

A metodologia experimental seguiu um protocolo análogo ao do refinamento anterior, conforme detalhado na Subseção 4.2.4: o modelo PIKAN foi treinado e avaliado no conjunto de validação para um intervalo de tamanhos de grade testados. As métricas quantitativas de desempenho (incluindo a média geral “*overall*”) foram registradas para cada valor de G .

A apresentação dos resultados são expostas na Figura 5.8. A análise quantitativa in-

Figura 5.8: Avaliação quantitativa do desempenho do modelo PIKAN em função do tamanho de grade (G) no conjunto de validação.



Fonte: Elaborado pelo autor (2025).

dica que o desempenho preditivo do modelo PIKAN se manteve estável e consistente em toda a faixa de tamanhos de grade testada. Essa estabilidade é evidenciada pela linha preta (“Overall”) nos gráficos de MAPE, MAE e RMSE, que apresenta pouca variação, oscilando entre 0.039 e 0.043 para o MAPE. A ausência de uma degradação significativa no desempenho para grades menores ou um ganho substancial para grades maiores é um achado importante, pois indica que a arquitetura PIKAN é robusta a variações moderadas na resolução da grade de splines, e que a complexidade do modelo pode ser controlada sem comprometer a precisão.

A escolha final do hiperparâmetro foi o tamanho de grade $G = 12$. Esta seleção foi justificada por representar um bom compromisso entre o desempenho preditivo e o custo computacional. O valor $G = 12$ permite que o modelo mantenha a alta capacidade preditiva observada, ao mesmo tempo que otimiza a eficiência computacional.

A configuração final do modelo PIKAN, com $s = 30$ e $G = 12$, foi então utilizada para a avaliação final de desempenho e eficiência computacional, cujos resultados são apresentados na Tabela 5.4 e discutidos nos próximos blocos desta seção.

Configuração do Treinamento Final

Para o treinamento do modelo inicial, foi utilizado a seguinte arquitetura PIKAN:

- **Camadas de entrada e saída:** Iguais ao modelo parametrizado. Dimensionadas de acordo com o número de características de entrada e variáveis alvo, respectivamente. Ou seja, 4 de entrada e 4 de saída.
- **Camadas intermediárias:** Duas camadas intermediárias com configurações 12 e 6, representando o número de nós somadores da primeira e segunda camada oculta,

respectivamente.

- **Grau polinomial:** $k = 3$, correspondendo a B-splines cúbicas.
- **Intervalo da grade:** $[0, 10]$, definindo o domínio de operação das funções splines.

O treinamento foi realizado utilizando, também, o otimizador Adam com taxa de aprendizado de $\eta = 10^{-3}$ e função de perda de dados MSE (*Mean Squared Error*).

A função de perda física utilizada é a mesma do modelo parametrizado, definida pela Equação 4.16. Todos os valores utilizados na perda física foram obtidas da mesma forma que no modelo não parametrizado, isto é, a partir da diferenciação automática do PyTorch e/ou pelo próprio RK2.

Assim como no modelo parametrizado, a PIKAN apresentou dificuldade na convergência do traçamento de raios nos pontos iniciais (para $t = 0$), portanto, também foi adotado a estratégia de acrescentar um erro quadrático médio à função de perda física para as amostras em $t = 0$.

Diferente do modelo parametrizado, os termos de regularização (entropia e L1) foram utilizados neste experimento. Os pesos utilizados foram:

$$\begin{aligned}\lambda &= 1 \times 10^{-3} \\ \lambda_{L1} &= 1 \times 10^{-3} \\ \lambda_{\text{ent}} &= 2\end{aligned}$$

Em que λ é o peso utilizado em todos os termos de regularização, λ_{L1} o peso utilizado apenas para a regularização L1, e λ_{ent} o peso utilizado apenas para a regularização de entropia.

As principais configurações adotadas foram:

- **Tamanho da grade:** $G = 12$. Por ser um modelo mais complexo, a hipótese inicial era que uma grade maior seria melhor.
- **Número de épocas:** 100, um número de épocas maior que o modelo parametrizado, pela mesma justificativa do hiperparâmetro anterior.
- **Taxa de aprendizagem inicial:** $\eta_0 = 10^{-3}$, sendo este um valor bastante presente na literatura como valor padrão.

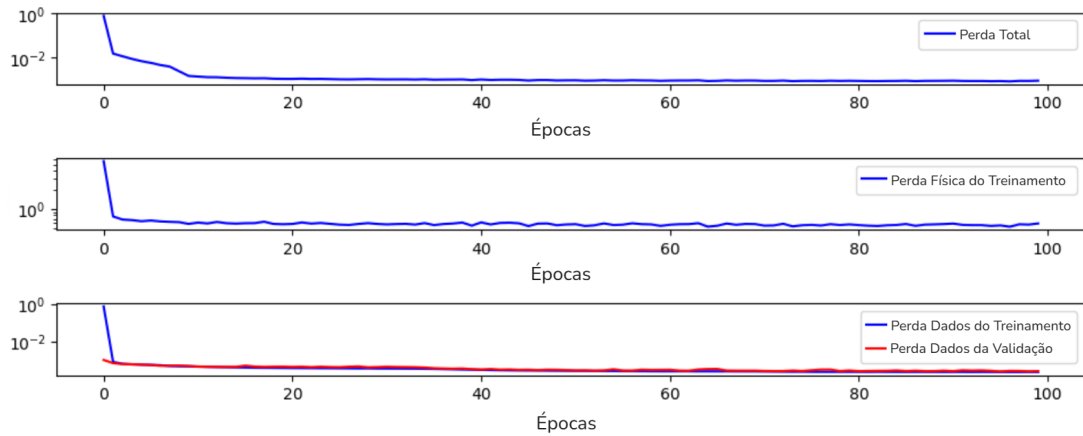
Os demais parâmetros do modelo foram mantidos idênticos aos utilizados no experimento anterior, incluindo a função de perda composta (dados + física), a técnica de *singularity avoiding* e a ponderação adaptativa da perda física.

Evolução do Treinamento

A Figura 5.9 apresenta a evolução das diferentes componentes da função de perda ao longo das 100 épocas de treinamento. Os gráficos mostram três aspectos fundamentais do processo de aprendizagem: a perda total de treinamento, a perda física de treinamento e a comparação entre as perdas de dados nos conjuntos de treinamento e validação.

A análise dos gráficos revela padrões no processo de aprendizagem parecidos com os obtidos no primeiro experimento:

Figura 5.9: Evolução das componentes da função de perda durante o treinamento do modelo PIKAN no modelo de velocidade do Marmousi. Painel superior: perda total de treinamento. Painel central: perda física de treinamento. Painel inferior: comparação entre as perdas de dados nos conjuntos de treinamento e validação.



Fonte: Elaborado pelo autor (2025).

1. **Perda total de treinamento:** Observa-se uma rápida diminuição nas primeiras épocas, estabilizando-se em torno de 10^{-2} , com pequenas movimentações nas épocas subsequentes.
2. **Perda física de treinamento:** Apresenta comportamento similar à perda total, com redução acentuada nas fases iniciais e estabilização rápida após a época 5.
3. **Perdas de dados:** As curvas de treinamento e validação mostram um padrão estável ao longo de todo o processo, com resultados bastante semelhantes entre elas. Essa semelhança sugere que o modelo apresenta um bom equilíbrio, sem sinais de sobreajuste ou subajuste, e mantém uma capacidade de generalização comparável tanto para os dados de treinamento quanto para os dados não vistos.

Um aspecto relevante, também observado no primeiro experimento, é a convergência das perdas de treinamento e validação, mesmo após 100 épocas. Esse comportamento indica que o aumento no tamanho da grade ($G = 12$) foi eficaz em mitigar o subajuste, um problema frequentemente encontrado em tarefas de maior complexidade.

Avaliação Quantitativa

Após o treinamento, o modelo foi avaliado no conjunto de teste, que contém trajetórias de raios com condições iniciais não vistas durante o treinamento ou validação. A Tabela 5.4 apresenta as métricas de desempenho obtidas para cada uma das quatro variáveis alvo.

Os resultados quantitativos demonstram resultados promissores:

- **Coefficiente de Determinação (R^2):** Valores acima de 0.98 para todas as variáveis alvo, indicando que o modelo explica muito bem a variância nos dados. Um ponto

	x	z	p_x	p_z
R^2	0.9999	0.9998	0.9886	0.9925
MAPE	0.061%	0.262%	0.995%	2.036%
MAE	3.29e−03	3.25e−03	4.57e−03	5.28e−03
MSE	2.08e−05	1.88e−05	3.52e−05	5.94e−05
RMSE	4.56e−03	4.33e−03	5.93e−03	7.71e−03

Tabela 5.4: Resultados obtidos para o traçamento de raios com PIKAN no modelo de velocidade do Marmousi.

de atenção é o p_x , que obteve o R^2 de 0.9886, aproximadamente, sendo o menor valor para este experimento.

- **Erro Percentual Absoluto Médio (MAPE):** Valores inferiores a 1% para todas as variáveis, exceto para a componente p_z do vetor de vagarosidade, se tornando, também, um ponto de atenção para os experimentos futuros.
- **Erro Absoluto Médio (MAE):** Valores na ordem de 10^{-3} para todas as variáveis, representando um erro médio de alguns metros, demonstrando alta precisão nas previsões.
- **Raiz do Erro Quadrático Médio (RMSE):** Valores consistentemente baixos, com o menor erro (4.56e−03) para z e o maior (7.71e−03) para p_z .

É notável que, apesar do pequeno aumento na complexidade da arquitetura, o modelo conseguiu capturar com alta precisão tanto as posições quanto as componentes do vetor de vagarosidade ao longo das trajetórias dos raios. Este resultado valida a eficácia da abordagem PIKAN para o problema do traçamento de raios sísmicos em cenários reais.

Avaliação Qualitativa

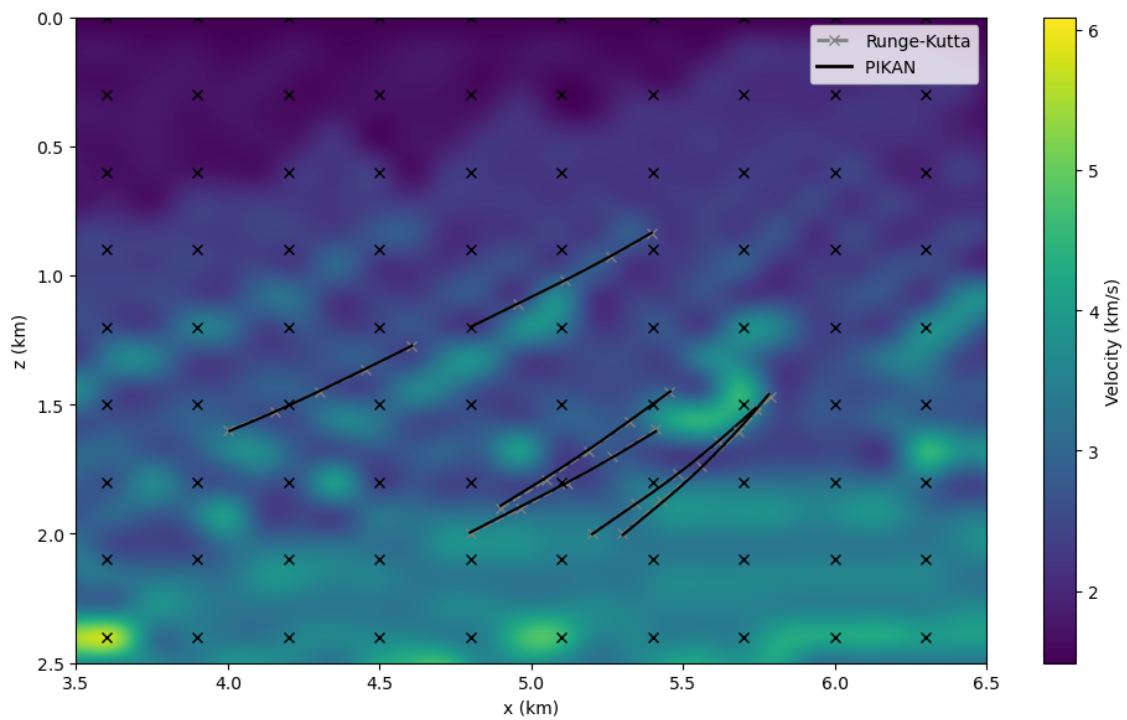
A Figura 5.10 apresenta a comparação das trajetórias previstas pelo modelo PIKAN com as trajetórias de referência calculadas pelo método de Runge-Kutta de segunda ordem, para um subconjunto representativo de raios no campo de velocidade de Marmousi.

A inspeção visual confirma o excelente desempenho do modelo PIKAN, com as trajetórias previstas muito próximas das trajetórias de referência. Além disso, o modelo PIKAN aplicado no modelo de velocidade do Marmousi também passa nos três critérios estabelecidos na Subseção 4.3.4, demonstrando que o modelo reproduz os valores numéricos com alta precisão e se ajustou às equações que modelam o fenômeno físico de propagação de onda sísmicas em meios heterogêneos, mesmo em um cenário mais complexo, mais próximo da realidade.

5.2.4 Comparação entre PIKAN e PINN tradicional

Nesta subseção, comparamos o desempenho da rede *Physics-Informed Kolmogorov-Arnold Network* (PIKAN) com uma PINN tradicional baseada em MLP, ambas aplicadas ao traçamento de raios sísmicos no modelo de velocidade Marmousi. O objetivo é isolar

Figura 5.10: Comparação visual entre trajetórias de raios sísmicos calculadas pelo método de Runge-Kutta de segunda ordem (linhas sólidas cinzas) e previstas pelo modelo PIKAN (linhas sólidas pretas). O campo de velocidade parametrizado é representado pelo gradiente de cores, com valores variando de 1,5 km/s (roxo) a 6 km/s (amarelo).



Fonte: Elaborado pelo autor (2025).

o efeito da arquitetura (KAN vs. MLP), mantendo constantes o conjunto de dados, a formulação da perda com informação física e a ordem de grandeza do número de parâmetros treináveis. Dessa forma, diferenças observadas podem ser atribuídas predominantemente à capacidade de representação de cada arquitetura.

Ambos os modelos recebem como entrada o mesmo vetor de variável e realiza a predição das mesmas variáveis de saída. A base de dados utilizada é a mesma do experimento anterior. Em ambos os casos, é utilizado a função custo definida na Equação 4.14. Outro ponto comum é o esquema de ponderação espacial da perda. Por fim, as duas redes são treinadas com o mesmo otimizador Adam (taxa de aprendizado inicial 10^{-2}) e agendamento via `ReduceLROnPlateau`, além de critérios de parada comparáveis, assegurando um regime de otimização equivalente.

Arquiteturas e número de parâmetros

A arquitetura PIKAN é implementada como uma KAN com largura

$$[\text{input_size}, 12, 6, \text{output_size}], \quad (5.25)$$

grade interna de base spline $G = 12$ e ordem $k = 3$. Essa configuração resulta em exatamente 3818 parâmetros treináveis.

Para uma comparação justa, foi projetada uma PINN tradicional baseada em MLP densa com número de parâmetros similar. A MLP possui quatro camadas ocultas de 36 neurônios, com ativação ReLU:

$$[\text{input_size}, 36, 36, 36, 36, \text{output_size}], \quad (5.26)$$

totalizando aproximadamente 4000 parâmetros. Portanto, ambas as arquiteturas têm quantidades de parâmetros treináveis comparáveis, garantindo que a comparação reflita diferenças arquiteturais e não mera capacidade numérica adicional.

Desempenho preditivo em teste

A Tabela 5.5 resume as métricas obtidas para os dois modelos no mesmo conjunto de teste. São reportados o coeficiente de determinação (R^2), o erro percentual absoluto médio (MAPE), o erro absoluto médio (MAE) e a raiz do erro quadrático médio (RMSE) para cada variável alvo.

Métrica	PIKAN (KAN)				PINN (MLP)			
	x	z	p_x	p_z	x	z	p_x	p_z
R^2	0.9999	0.9998	0.9886	0.9925	0.9998	0.9994	0.9714	0.9869
MAPE	0.061%	0.262%	0.995%	2.036%	0.107%	0.475%	1.597%	2.918%
MAE	3.29e−03	3.25e−03	4.57e−03	5.28e−03	5.82e−03	5.89e−03	7.37e−03	7.80e−03
RMSE	4.56e−03	4.33e−03	5.93e−03	7.71e−03	7.84e−03	7.71e−03	9.40e−03	1.02e−02

Tabela 5.5: Comparação quantitativa entre PIKAN (KAN) e PINN tradicional (MLP) no conjunto de teste do Marmousi.

Os resultados indicam que ambas as arquiteturas atingem excelente aderência às trajetórias de referência, com R^2 muito próximo de 1 em todas as variáveis. Entretanto, a PIKAN apresenta erros sistematicamente menores em todas as métricas e saídas, com ganhos mais expressivos em p_x e p_z , que são quantidades diretamente controladas pelos gradientes do campo de velocidade. Isso sugere que a representação funcional das KANs favorece a captura de variações sutis do meio e de suas derivadas, produzindo trajetórias fisicamente mais consistentes em um cenário altamente heterogêneo como o Marmousi.

Custo computacional de inferência

O tempo de execução para o traçamento de 323 trajetórias no conjunto de teste foi comparado entre o método numérico de referência e as duas redes neurais pré-treinadas. O método de Runge–Kutta de segunda ordem demanda aproximadamente 16 s por experimento, excluindo o tempo necessário para o pré-processamento dos dados. Em contraste, a inferência da PIKAN leva em torno de 66.6 ± 43.7 ms, enquanto a PINN é ainda mais rápida, com aproximadamente 18.9 ± 21.6 ms nas mesmas condições. Essa diferença é esperada, pois a PINN baseada em MLP utiliza apenas operações matriciais densas e ativações elementares, altamente otimizadas em bibliotecas de aprendizado profundo, como o Pytorch, ao passo que a PIKAN incorpora camadas KAN com avaliação de B-splines e combinações funcionais unidimensionais, o que envolve maior custo por *forward pass*.

Apesar disso, ambas as arquiteturas fornecem aceleração de várias ordens de magnitude em relação ao método numérico, reduzindo o tempo de traçamento de segundos para dezenas de milissegundos. Além disso, os tempos de inferência de ambos os modelos são suficientemente pequenos para aplicações reais: a diferença absoluta entre PIKAN e PINN (da ordem de poucas dezenas de milissegundos para centenas de trajetórias) não impactaria substancialmente um cenário operativo típico, especialmente quando comparada ao ganho massivo obtido frente ao método Runge-Kutta. Nesse contexto, o leve sobrecusto computacional da PIKAN tende a ser plenamente aceitável e facilmente compensado por sua maior capacidade preditiva e interpretabilidade da rede.

Capítulo 6

Conclusão

Esta dissertação propôs, implementou e validou uma nova metodologia para o traçamento de raios sísmicos em meios heterogêneos, fundamentada em Redes de Kolmogorov-Arnold com Informação Física (PIKANs). Partiu-se da hipótese de que a combinação entre (i) a alta capacidade intrínseca de aproximação funcional das redes de Kolmogorov-Arnold (KANs) e (ii) a inclusão explícita das restrições físicas definidas pelas equações canônicas de Hamilton poderia resultar um traçador neural mais preciso, mais estável e com melhor generalização em comparação com as PINNs tradicionais baseadas em MLP. Os experimentos realizados ao longo do trabalho sustentam essa hipótese: a PIKAN demonstrou consistentemente desempenho superior, especialmente nas variáveis dinâmicas mais sensíveis aos gradientes do campo de velocidade. Além disso, manteve um custo de inferência consideravelmente menor frente ao método numérico de referência e desprezível frente às PINNs tradicionais.

6.1 Síntese das contribuições

As principais contribuições desta dissertação para a geofísica computacional e para o aprendizado de máquina com informação física podem ser sintetizadas em:

1. **Framework PIKAN para traçamento de raios sísmicos.** Foi desenvolvido um framework completo que formula o problema de traçamento de raios como um sistema de EDOs hamiltonianas e o resolve por meio de redes PIKAN. Trata-se, ao melhor do nosso conhecimento, da primeira aplicação de KANs com informação física a esse problema, abrindo uma nova linha de investigação para métodos neurais em modelagem geofísica.
2. **Generalização para um conjunto amplo de condições iniciais.** Ao contrário de muitas PINNs voltadas a trajetórias específicas, a PIKAN foi treinado para aprender um operador dinâmico condicionado por (x_0, z_0, θ_0) , demonstrando excelente capacidade de generalização para condições iniciais não vistas. Essa característica é fundamental em aplicações reais de imageamento, nas quais milhares de raios com diferentes disparos precisam ser avaliados.
3. **Função de perda composta, adaptativa e com informações físicas.** Implementou-se uma função de perda híbrida que equilibra supervisão por dados e penalização

física via diferenciação automática das equações canônicas. Além disso, empregou-se ponderação espacial adaptativa do termo físico, mitigando vieses de amostragem e impondo consistência também em regiões menos densas do domínio.

4. **Validação em cenários sintéticos de complexidade crescente.** O método foi validado em um modelo parametrizado linear e no Marmousi, mostrando alta acurácia em ambos. No caso mais complexo, a PIKAN obteve resultados comparáveis ou superiores ao Runge–Kutta de segunda ordem, com a vantagem de fornecer uma solução contínua, diferenciável e consultável em tempo real após o treinamento.

6.2 Limitações do trabalho

Apesar dos resultados positivos, algumas limitações relacionadas ao escopo desta dissertação devem ser destacadas:

- **Dimensionalidade.** O estudo foi conduzido em 2D. A extensão para 3D é natural, porém envolve aumento substancial na complexidade geométrica, na dimensionalidade de entrada/saída e no custo computacional, exigindo adaptações de arquitetura e de geração de dados.
- **Custo de treinamento.** Embora a inferência da PIKAN seja extremamente rápida, o treinamento em modelos mais complexos (como o Marmousi) permanece computacionalmente custoso, característica compartilhada por PINNs e métodos neurais informados por física em geral.
- **Retreinamento para Novos Modelos de Velocidade.** O modelo PIKAN é treinado para um modelo de velocidade específico (e.g., Marmousi). A aplicação a um novo modelo de velocidade requer o retreinamento completo da rede, o que pode ser computacionalmente intensivo e limita a aplicação imediata a novos cenários geológicos.

6.3 Trabalhos futuros

A partir das contribuições e limitações levantadas, apresentam-se direções para pesquisas futuras:

1. **Aplicação a modelos do pré-sal brasileiro.** Avaliar a PIKAN em modelos típicos do pré-sal, caracterizados por alto contraste de velocidades, geometrias complexas e presença de sal. Esse cenário representa um teste de robustez com forte relevância industrial e acadêmica.
2. **Traçamento orientado a tarefas geofísicas específicas.** Integrar a PIKAN em aplicações de rotina geofísica, como cálculo de tempos de trânsito, migração e conexão tiro-receptor em modelos de aquisição sísmica, explorando a consulta rápida de trajetórias condicionadas pelas fontes e receptores.
3. **Controle de qualidade em problemas inversos.** Investigar o impacto das trajetórias preditas pela PIKAN em tarefas inversas, como tomografia sísmica. Com-

parações diretas com métodos numéricos podem quantificar ganhos (ou limites) na qualidade de inversão.

4. **Mapeamento do resíduo hamiltoniano.** Construir mapas de calor do máximo resíduo das equações de Hamilton no domínio, avaliando a aderência física local da PIKAN. A comparação com mapas gerados pelo Runge-Kutta forneceria uma importante ferramenta diagnóstica objetiva.
5. **Dependência do erro em função das condições iniciais.** Ao contrário de muitas PINNs voltadas ao aprendizado de trajetórias específicas, a PIKAN foi treinado para aprender um operador dinâmico condicionado por (x_0, z_0, θ_0) , demonstrando excelente capacidade de generalização para condições iniciais não vistas. Essa característica é fundamental em aplicações reais de imageamento, nas quais milhares de raios com diferentes direções de disparos precisam ser avaliados.
6. **Extensão para a equação Eikonal.** Adaptar o framework PIKAN para resolver a equação Eikonal, explorando a capacidade de KANs informadas por física em EDPs de primeira ordem e ampliando a classe de problemas geofísicos tratáveis.

6.4 Considerações finais

Em resumo, esta dissertação fundamentou o uso de redes de Kolmogorov-Arnold com Informação Física no traçamento de raios sísmicos, evidenciando melhorias significativas em precisão, generalização e consistência física em relação às abordagens tradicionais baseadas em PINNs, em condições experimentais controladas e com número de parâmetros semelhante. Os resultados sugerem que a PIKAN é uma opção promissora para acelerar problemas diretos em geofísica computacional, particularmente em domínios mais complexos nos quais os gradientes do meio têm um papel fundamental. As extensões propostas têm como objetivo fortalecer essa abordagem como um instrumento sólido e eficaz para a modelagem da propagação de ondas em meios geológicos realistas.

Referências Bibliográficas

- Badouel, Didier, Kadi Bouatouch & Thierry Priol (1990), Ray tracing on distributed memory parallel computers : strategies for distributing computations and data, Research Report RR-1163, INRIA.
URL: <https://inria.hal.science/inria-00075395>
- Baydin, Atilim Gunes, Barak A. Pearlmutter, Alexey Andreyevich Radul & Jeffrey Mark Siskind (2018), ‘Automatic differentiation in machine learning: a survey’, *Journal of Machine Learning Research* **18**(153), 1–43.
URL: <http://jmlr.org/papers/v18/17-468.html>
- Bergen, Karianne J., Paul A. Johnson, Maarten V. de Hoop & Gregory C. Beroza (2019), ‘Machine learning for data-driven discovery in solid earth geoscience’, *Science* **363**(6433), eaau0323.
URL: <https://www.science.org/doi/abs/10.1126/science.aau0323>
- Billette, Frédéric & Gilles Lambaré (1998), ‘Velocity macro-model estimation from seismic reflection data by stereotomography’, *Geophysical Journal International* **135**(2), 671–690.
URL: <https://doi.org/10.1046/j.1365-246X.1998.00632.x>
- bin Waheed, Umair, Ehsan Haghighat, Tariq Alkhalifah, Chao Song & Qi Hao (2021), ‘Pinneik: Eikonal solution using physics-informed neural networks’, *Computers & Geosciences* **155**, 104833.
URL: <https://www.sciencedirect.com/science/article/pii/S009830042100131X>
- Bishop, Christopher M. (2006), *Pattern Recognition and Machine Learning*, Information Science and Statistics, 1ª edição, Springer New York, NY.
- Botchkarev, Alexei (2019), ‘A new typology design of performance metrics to measure errors in machine learning regression algorithms’, *Interdisciplinary Journal of Information, Knowledge, and Management* **14**, 045–076.
URL: <http://dx.doi.org/10.28945/4184>
- Briand, Thibaud & Pascal Monasse (2018), ‘Theory and Practice of Image B-Spline Interpolation’, *Image Processing On Line* **8**, 99–141.
URL: <https://enpc.hal.science/hal-01846912>
- Carcione, José M., Gérard C. Herman & A. P. E. ten Kroode (2002), ‘Seismic modeling’, *Geophysics* **67**(4), 1304–1325.
URL: <https://doi.org/10.1190/1.1500393>

- Červený, Vlastislav (1989), 'Ray tracing in factorized anisotropic inhomogeneous media', *Geophysical Journal International* **99**(1), 91–100.
- Červený, Vlastislav (2001), *Seismic Ray Theory*, Cambridge University Press, Cambridge.
- Chai, T. & R. R. Draxler (2014), 'Root mean square error (rmse) or mean absolute error (mae)? – arguments against avoiding rmse in the literature', *Geoscientific Model Development* **7**(3), 1247–1250.
URL: <https://gmd.copernicus.org/articles/7/1247/2014/>
- Cuomo, Salvatore, Vincenzo S. Di Cola, Francesco Giampaolo, Riccardo de Maio, Rosario Monti & Fiammetta Piccialli (2022), 'Scientific machine learning through physics-informed neural networks: Where we are and what's next', *Journal of Scientific Computing* **92**(88), 1–35.
URL: <https://doi.org/10.1007/s10915-022-01939-z>
- de Boor, Carl (1978), *A Practical Guide to Splines*, Applied Mathematical Sciences, Springer.
- Dhiman, Vikas (2024), 'Kan: Kolmogorov–arnold networks: A review', *Retrieved from* .
- Draper, Norman Richard & Harry Smith (1966), *Applied regression analysis*, Wiley series in probability and mathematical statistics, Wiley, New York [u.a.].
- Duarte, Diogo H. G., Ramon C. F. Araújo, Paulo D. S. de Lima, João M. de Araújo, Tiago Barros & Gilberto Corso (2023), Ray tracing and physics-informed neural networks. Apresentado no 18º Congresso Internacional da Sociedade Brasileira de Geofísica & Expogef, Rio de Janeiro, Brasil, 16–19 de Outubro de 2023.
- Eivazi, Hamidreza, Mojtaba Tahani, Philipp Schlatter & Ricardo Vinuesa (2022), 'Physics-informed neural networks for solving reynolds-averaged navier–stokes equations', *Physics of Fluids* **34**(7), 075117.
URL: <https://doi.org/10.1063/5.0095270>
- Farra, V. & R. Madariaga (1988), 'Non-linear reflection tomography', *Geophysical Journal International* **95**(1), 135–147.
URL: <https://doi.org/10.1111/j.1365-246X.1988.tb00456.x>
- Gao, Bo, Ruoxia Yao & Yan Li (2025), 'Physics-informed neural networks with adaptive loss weighting algorithm for solving partial differential equations', *Computers & Mathematics with Applications* **181**, 216–227.
URL: <https://www.sciencedirect.com/science/article/pii/S0898122125000069>
- Geoazur (2025), 'Marmousi model'. Acessado em 17/01/2025.
URL: <https://www.geoazur.fr/WIND/bin/view/Main/Data/Marmousi>

- Goodfellow, Ian, Yoshua Bengio & Aaron Courville (2016), *Deep Learning*, MIT Press, Cambridge, MA.
URL: <https://www.deeplearningBook.org>
- Hadj Kilani, Bochra (2024), ‘Kolmogorov-Arnold Networks: Key Developments and Uses’, *Qeios* .
URL: <https://hal.science/hal-04663521>
- Hastie, Trevor, Robert Tibshirani & Jerome Friedman (2009), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2ª edição, Springer, New York.
- Hyndman, Rob J. & Anne B. Koehler (2006), ‘Another look at measures of forecast accuracy’, *International Journal of Forecasting* **22**(4), 679–688.
URL: <https://www.sciencedirect.com/science/article/pii/S0169207006000239>
- James, Gareth, Daniela Witten, Trevor Hastie & Robert Tibshirani (2013), *An Introduction to Statistical Learning: with Applications in R*, Springer, New York.
- Jaroš, Milan, Lubomír Říha, Petr Strakoš & Matěj Špet’ko (2021), ‘Gpu accelerated path tracing of massive scenes’, *ACM Trans. Graph.* **40**(2).
URL: <https://doi.org/10.1145/3447807>
- Julian, Bruce R. (1970), Ray tracing in arbitrarily heterogeneous media, Relatório Técnico Technical Note 1970-45, Massachusetts Institute of Technology, Lincoln Laboratory, Lexington, MA. DTIC Accession No. AD0720795.
URL: <https://apps.dtic.mil/sti/html/tr/AD0720795>
- Karniadakis, George Em, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang & Liu Yang (2021), ‘Physics-informed machine learning’, *Nature Reviews Physics* **3**(6), 422–440.
URL: <https://doi.org/10.1038/s42254-021-00314-5>
- Kaufman, Shachar, Saharon Rosset, Claudia Perlich & Ori Stitelman (2012), ‘Leakage in data mining: Formulation, detection, and avoidance’, *ACM Trans. Knowl. Discov. Data* **6**(4).
URL: <https://doi.org/10.1145/2382577.2382579>
- Kingma, Diederik P. & Jimmy Ba (2017), ‘Adam: A method for stochastic optimization’.
URL: <https://arxiv.org/abs/1412.6980>
- Koketsu, Kazuki & Shutaro Sekine (1998), ‘Pseudo-bending method for three-dimensional seismic ray tracing in a spherical earth with discontinuities’, *Geophysical Journal International* **132**(2), 339–346.
URL: <https://doi.org/10.1046/j.1365-246x.1998.00427.x>
- Kolmogorov, A.N. (1957), ‘On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition’, *Doklady Akademii Nauk SSSR* **144**, 679–681.

- Krishnapriyan, Aditi, Amir Gholami, Shandian Zhe, Robert Kirby & Michael W Mahoney (2021), Characterizing possible failure modes in physics-informed neural networks, *em* M.Ranzato, A.Beygelzimer, Y.Dauphin, P. S.Liang & J. W.Vaughan, eds., ‘Advances in Neural Information Processing Systems’, Vol. 34, Curran Associates, Inc., pp. 26548–26560.
URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/df438e5206f31600e6ae4af72f2725f1-Paper.pdf
- Lambaré, G., P. S. Lucio & A. Hanyga (1996), ‘Two-dimensional multivalued traveltime and amplitude maps by uniform sampling of a ray field’, *Geophysical Journal International* **125**(2), 584–598.
URL: <https://doi.org/10.1111/j.1365-246X.1996.tb00021.x>
- Liu, Ziming, Pingchuan Ma, Yixuan Wang, Wojciech Matusik & Max Tegmark (2024), ‘Kan 2.0: Kolmogorov-arnold networks meet science’.
URL: <https://arxiv.org/abs/2408.10205>
- Liu, Ziming, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou & Max Tegmark (2025), ‘Kan: Kolmogorov-arnold networks’.
URL: <https://arxiv.org/abs/2404.19756>
- Loshchilov, Ilya & Frank Hutter (2017), ‘Sgdr: Stochastic gradient descent with warm restarts’.
URL: <https://arxiv.org/abs/1608.03983>
- Makridakis, Spyros (1993), ‘Accuracy measures: theoretical and practical concerns’, *International Journal of Forecasting* **9**(4), 527–529.
URL: <https://www.sciencedirect.com/science/article/pii/0169207093900793>
- Margrave, Gary F (2003), ‘Numerical methods of exploration seismology with algorithms in MATLAB’, *Course Notes, CREWES Project, University of Calgary*.
- Margrave, Gary F. & Michael P. Lamoureux (2019), *Numerical Methods of Exploration Seismology: With Algorithms in MATLAB*, Cambridge University Press.
- Moseley, Ben, Andrew Markham & Tarje Nissen-Meyer (2020a), ‘Solving the wave equation with physics-informed deep learning’.
URL: <https://arxiv.org/abs/2006.11894>
- Moseley, Ben, Andrew Markham & Tarje Nissen-Meyer (2020b), ‘Solving the wave equation with physics-informed deep learning’.
URL: <https://arxiv.org/abs/2006.11894>
- Müller, Johannes & Marius Zeinhofer (2023), Achieving high accuracy with PINNs via energy natural gradient descent, *em* A.Krause, E.Brunskill, K.Cho, B.Engelhardt, S.Sabato & J.Scarlett, eds., ‘Proceedings of the 40th International Conference on Machine Learning’, Vol. 202 de *Proceedings of Machine Learning Research*,

- PMLR, pp. 25471–25485.
URL: <https://proceedings.mlr.press/v202/muller23b.html>
- Muniz, João Paulo Castanheira (2016), Estudos de iluminação sísmica por traçado de raios, Tese de doutorado, Universidade Federal Fluminense, Niterói, Brasil.
URL: https://geofisica.uff.br/sites/default/files/projetofinal/tcc_joaomuniz.pdf
- Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai & Soumith Chintala (2019), Pytorch: An imperative style, high-performance deep learning library, *em* H.Wallach, H.Larochelle, A.Beygelzimer, F.d'Alché-Buc, E.Fox & R.Garnett, eds., 'Advances in Neural Information Processing Systems', Vol. 32, Curran Associates, Inc.
URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf
- Patra, Subhajit, Sonali Panda, Bikram Keshari Parida, Mahima Arya, Kurt Jacobs, Denys I. Bondar & Abhijit Sen (2024), 'Physics informed kolmogorov-arnold neural networks for dynamical analysis via efficient-kan and wav-kan'.
URL: <https://arxiv.org/abs/2407.18373>
- Pereyra, V. (1992), 'Two-point ray tracing in general 3d media', *Geophysical Prospecting* **40**(3), 267–287.
URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2478.1992.tb00375.x>
- Pineau, Joelle, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox & Hugo Larochelle (2021), 'Improving reproducibility in machine learning research(a report from the neurips 2019 reproducibility program)', *Journal of Machine Learning Research* **22**(164), 1–20.
URL: <http://jmlr.org/papers/v22/20-303.html>
- Raissi, M., P. Perdikaris & G.E. Karniadakis (2019), 'Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations', *Journal of Computational Physics* **378**, 686–707.
URL: <https://www.sciencedirect.com/science/article/pii/S0021999118307125>
- Rawlinson, N., J. Hauser & M. Sambridge (2008), Seismic ray tracing and wavefront tracking in laterally heterogeneous media, Vol. 49 de *Advances in Geophysics*, Elsevier, pp. 203–273.
URL: <https://www.sciencedirect.com/science/article/pii/S0065268707490033>
- Rawlinson, N. & M. Sambridge (2004), 'Wave front evolution in strongly heterogeneous layered media using the fast marching method', *Geophysical Journal International*

156(3), 631–647.

URL: <https://doi.org/10.1111/j.1365-246X.2004.02153.x>

Romero, Juan, Wolfgang Heidrich, Nick Luiken & Matteo Ravasi (2024), ‘Seismic reservoir characterization with implicit neural representations’, *Journal of Geophysical Research: Machine Learning and Computation* **1**(3), e2024JH000275. e2024JH000275 2024JH000275.

URL: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2024JH000275>

Sheriff, Robert E (2002), *Encyclopedic Dictionary of Applied Geophysics*, 4th^a edição, Society of Exploration Geophysicists, Tulsa, OK.

Shuai, Hang & Fangxing Li (2025), ‘Physics-informed kolmogorov-arnold networks for power system dynamics’, *IEEE Open Access Journal of Power and Energy* **12**, 46–58.

Shukla, Khemraj, Juan Diego Toscano, Zhicheng Wang, Zongren Zou & George Em Karniadakis (2024), ‘A comprehensive and fair comparison between mlp and kan representations for differential equations and operator networks’, *Computer Methods in Applied Mechanics and Engineering* **431**, 117290.

URL: <https://www.sciencedirect.com/science/article/pii/S0045782524005462>

Shulga, Dmytro, Oleksii Morozov, Volker Roth, Felix Friedrich & Patrick Hunziker (2019), ‘Tensor b-spline numerical methods for pdes: a high-performance alternative to fem’.

URL: <https://arxiv.org/abs/1904.03057>

Smith, Jonathan D., Kamyar Azizzadenesheli & Zachary E. Ross (2021), ‘Eikonet: Solving the eikonal equation with deep neural networks’, *IEEE Transactions on Geoscience and Remote Sensing* **59**(12), 10685–10696.

Smith, Leslie N. (2018), ‘A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay’.

URL: <https://arxiv.org/abs/1803.09820>

Tibshirani, Robert (2018), ‘Regression shrinkage and selection via the lasso’, *Journal of the Royal Statistical Society: Series B (Methodological)* **58**(1), 267–288.

URL: <https://doi.org/10.1111/j.2517-6161.1996.tb02080.x>

Unser, M. (1999), ‘Splines: a perfect fit for signal and image processing’, *IEEE Signal Processing Magazine* **16**(6), 22–38.

Unser, M., A. Aldroubi & M. Eden (1993a), ‘B-spline signal processing. ii. efficiency design and applications’, *IEEE Transactions on Signal Processing* **41**(2), 834–848.

Unser, Michael, Akram Aldroubi & Michael Eden (1993b), ‘B-spline signal processing: Part i—theory’, *IEEE Transactions on Signal Processing* **41**(2), 821–833.

- Vesnaver, Aldo L. (1996), 'Ray tracing based on fermat's principle in irregular grids1', *Geophysical Prospecting* **44**(5), 741–760.
URL: <https://www.earthdoc.org/content/journals/10.1111/j.1365-2478.1996.tb00172.x>
- Virieux, J. & S. Operto (2009), 'An overview of full-waveform inversion in exploration geophysics', *Geophysics* **74**(6), WCC1–WCC26.
URL: <https://doi.org/10.1190/1.3238367>
- Virieux, Jean, Veronique Farra & Raul Madariaga (1988), 'Ray tracing for earthquake location in laterally heterogeneous media', *Journal of Geophysical Research: Solid Earth* **93**(B6), 6585–6599.
URL: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/JB093iB06p06585>
- Wang, Sifan, Shyam Sankaran & Paris Perdikaris (2022), 'Respecting causality is all you need for training physics-informed neural networks'.
URL: <https://arxiv.org/abs/2203.07404>
- Wang, Sifan, Yujun Teng & Paris Perdikaris (2021), 'Understanding and mitigating gradient flow pathologies in physics-informed neural networks', *SIAM Journal on Scientific Computing* **43**(5), A3055–A3081.
URL: <https://doi.org/10.1137/20M1318043>
- Wang, Yanghua (2013), 'Seismic ray tracing in anisotropic media: A modified newton algorithm for solving highly nonlinear systems', *Geophysics* **79**(1), T1–T7.
URL: <https://doi.org/10.1190/geo2013-0110.1>
- Wang, Yizheng, Jia Sun, Jinshuai Bai, Cosmin Anitescu, Mohammad Sadegh Eshaghi, Xiaoying Zhuang, Timon Rabczuk & Yinghua Liu (2025), 'Kolmogorov–arnold-informed neural network: A physics-informed deep learning framework for solving forward and inverse problems based on kolmogorov–arnold networks', *Computer Methods in Applied Mechanics and Engineering* **433**, 117518.
URL: <https://www.sciencedirect.com/science/article/pii/S0045782524007722>
- Wes McKinney (2010), Data Structures for Statistical Computing in Python, *em* Stéfan van der Walt & Jarrod Millman, eds., 'Proceedings of the 9th Python in Science Conference', pp. 56 – 61.
- Willmott, C. J. & K. Matsuura (2005), 'Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance', *Climate Research* **30**, 79–82.
URL: <https://www.int-res.com/abstracts/cr/v30/cr030079>
- Wilson, Greg, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, Kathryn D. Huff, Ian M. Mitchell, Mark D. Plumbly, Ben Waugh, Ethan P. White & Paul Wilson (2014), 'Best practices for scientific computing', *PLOS Biology* **12**(1), 1–7.
URL: <https://doi.org/10.1371/journal.pbio.1001745>

- Xiang, Zixue, Wei Peng, Xu Liu & Wen Yao (2022), ‘Self-adaptive loss balanced physics–informed neural networks’, *Neurocomputing* **496**, 11–34.
URL: <https://www.sciencedirect.com/science/article/pii/S092523122200546X>
- Yamak, Peter Tettey, Yujian Li, Ting Zhang & Muhammad Salman Pathan (2025), ‘Kolmogorov-arnold networks for time series forecasting: a comprehensive review’, *Cluster Computing* **28**(14).
URL: <https://doi.org/10.1007/s10586-025-05574-9>
- Yilmaz, Öz (2001), *Seismic Data Analysis: Processing, Inversion, and Interpretation of Seismic Data*, Society of Exploration Geophysicists.
URL: <https://doi.org/10.1190/1.9781560801580>
- Zelt, C. A. & R. M. Ellis (1988), ‘Practical and efficient ray tracing in two-dimensional media for rapid traveltimes and amplitude forward modelling’.
- Zhang, Minghui, Tao Xu, Zhiming Bai, Youshan Liu, Jue Hou & Guiping Yu (2017), ‘Ray tracing of turning wave in elliptically anisotropic media with an irregular surface’, *Earthquake Science* **30**(5), 219–228.
URL: <https://doi.org/10.1007/s11589-017-0192-5>
- Zhang, Zhaoyang, Qingwang Wang, Yinxing Zhang, Tao Shen & Weiyi Zhang (2025), ‘Physics–informed neural networks with hybrid kolmogorov-arnold network and augmented lagrangian function for solving partial differential equations’, *Scientific Reports* **15**(1).
URL: <https://doi.org/10.1038/s41598-025-92900-1>
- Zhu, Lijun, Zhigang Peng, James McClellan, Chenyu Li, Dongdong Yao, Zefeng Li & Lihua Fang (2019), ‘Deep learning for seismic phase detection and picking in the aftershock zone of 2008 mw7.9 wenchuan earthquake’, *Physics of the Earth and Planetary Interiors* **293**, 106261.
URL: <https://www.sciencedirect.com/science/article/pii/S0031920118301407>
- Červený, V. (1989), ‘Ray tracing in factorized anisotropic inhomogeneous media’, *Geophysical Journal International* **99**(1), 91–100.
URL: <https://doi.org/10.1111/j.1365-246X.1989.tb02017.x>